



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**“PROPUESTA DE UN MARCO DE REFERENCIA PARA FOMENTAR
LAS HABILIDADES SUAVES EN LA ENSEÑANZA DEL
PARADIGMA DE DESARROLLO GLOBAL DE SOFTWARE A NIVEL
LICENCIATURA”**

TESIS

**PARA OBTENER EL TÍTULO DE
MAESTRO EN TECNOLOGÍAS DE CÓMPUTO APLICADO**

PRESENTA

L. I. ANDRÉS LEÓN PÉREZ

**DIRECTOR DE TESIS
DR. IVÁN ANTONIO GARCÍA PACHECO**

HUAJUAPAN DE LEÓN, OAX., OCTUBRE DE 2019

**Esta tesis fue presentada el 11 de octubre de 2019,
ante los siguientes sinodales:**

Dra. Carla Leninca Pacheco Agüero
Dr. Carlos Alberto Fernández y Fernández
Dr. José Aníbal Arias Aguilar
M.I.S. Hugo Enrique Martínez Cortés

Director de tesis:

Dr. Iván Antonio García Pacheco

Dedicatoria

Muy en especial para una hermosa niña que ha llegado a iluminar nuestras vidas. Romi, tu presencia en nuestras vidas nos alienta a seguir adelante. Que este trabajo de tesis sirva de motivación para ti en un futuro no muy lejano para luchar, esforzarte, y alcanzar las metas que te propongas.

Para ti Luisa, que siempre estás conmigo, me apoyas, y nunca me dejas solo, tu amor y alegría me inspiran para seguir adelante.

A mis padres Doroteo y Carmela, su vida y forma de ser siempre tan amables, buenas personas, y excelentes padres me han inspirado para no rendirme y siempre dar lo mejor de mí en mis diferentes actividades.

A mis hermanos Javier y Ana Delia que siempre y a la distancia están conmigo apoyándome.

A mis sobrinos Ariadna Sarahí e Iker Daniel, espero también pueda este trabajo motivarlos a alcanzar sus metas. Gracias por enseñarme a ser tío.

Agradecimientos

De manera especial a mi director de tesis, Dr. Iván Antonio García Pacheco, por el tiempo dedicado a la revisión de este trabajo. Es un gran maestro, he aprendido mucho de usted no solo en aspectos técnicos y conocimientos, sino también en valores. Gracias por los consejos y el apoyo que recibí de parte de usted a lo largo de mis estudios de maestría.

A las universidades, profesores, estudiantes, y amigos que me apoyaron en la realización del caso de estudio que permitió validar este trabajo, gracias.

A mis profesores y a la Universidad Tecnológica de la Mixteca por permitirme formarme en algo que tanto me apasiona.

A mis sinodales Dra. Carla Leninca Pacheco Agüero, Dr. Carlos A. Fernández y Fernández, M. I. S. Hugo Enrique Martínez, y Dr. José Aníbal Arias Aguilar, gracias por los consejos y tiempo dedicado a la revisión de este trabajo.

A Luisa, por estar siempre a mi lado, soportar mis malos ratos y mi pésimo humor cuando las cosas no van del todo bien, por motivarme a seguir adelante con tu amor, apoyo, comprensión y ternura, gracias.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por brindarme el apoyo económico durante el periodo que abarcaron mis estudios de maestría, sin el cual hubiera sido imposible alcanzar tal objetivo.

Índice

Dedicatoria	v
Agradecimientos.....	vii
Índice	ix
Lista de tablas	xiii
Lista de figuras	xv
Resumen	xix
Abstract	xxi
1. Introducción y motivaciones	1
1.1. Contexto del problema.....	1
1.2. Importancia del problema.....	5
1.3. Necesidad de solución	8
1.4. Delimitaciones y limitaciones de la investigación	11
1.5. Hipótesis de la tesis	11
1.6. Objetivos de la tesis.....	11
1.6.1. Objetivo general	12
1.6.2. Objetivos secundarios.....	12
1.7. Aproximación metodológica para la solución.....	12
1.7.1. Diseño del marco de referencia	12
1.7.2. Diseño y construcción de la plataforma tecnológica.....	14
1.7.3. Experimentación.....	15
1.8. Estructura de la tesis	16
2. Marco teórico	17
2.1. Aspectos relevantes sobre la enseñanza de GSD en entornos universitarios	22
2.1.1. Los retos y desafíos en la educación de GSD.....	24
2.1.2. La enseñanza del GSD y la gestión de proyectos.....	29
2.2. El enfoque pedagógico en la implementación del GSD	31
2.2.1. Estado del arte	35
2.2.2. iBistro: Un entorno de aprendizaje para la construcción del conocimiento en los cursos de Ingeniería de Software de Sistemas Distribuidos.....	36

2.2.2.1. Enfoque de enseñanza.....	36
2.2.2.2. Resultados y conclusiones	39
2.2.3. Un aula virtual y global para enseñar Ingeniería Global de Software: Un proyecto de enseñanza de equipos mongol-alemanes	40
2.2.3.1. Enfoque de enseñanza.....	40
2.2.3.2. Resultados y conclusiones	42
2.2.4. Una plataforma para la enseñanza del desarrollo de software distribuido	42
2.2.4.1. Enfoque de enseñanza.....	42
2.2.4.2. Resultados y conclusiones	44
2.2.5. GSDGame: Un juego serio para adquirir las competencias necesarias en el GSD	44
2.2.5.1. Enfoque de enseñanza.....	44
2.2.5.2. Resultados y conclusiones	47
2.2.6. Enseñando un curso de desarrollo global de software: Experiencias de los estudiantes utilizando la simulación de ejercicios en el aula	47
2.2.6.1. Enfoque de enseñanza.....	47
2.2.6.2. Resultados y conclusiones	48
2.2.7. Consideraciones finales sobre el estado del arte.....	49
3. Desarrollo de la solución	53
3.1. Diseño del marco de referencia	53
3.1.1. Análisis de necesidades y filosofía educativa.....	56
3.1.1.1. Análisis de necesidades	56
3.1.1.2. Filosofía educativa	59
3.1.2. Análisis de modelos curriculares	60
3.1.3. Definición de objetivos y resultados.....	62
3.1.3.1. Atributos de los estudiantes	63
3.1.3.2. Objetivos del marco de referencia	63
3.1.3.3. Resultados de aprendizaje.....	64
3.1.4. Organización y estructura del programa.....	64
3.1.4.1. Alcance	64
3.1.4.2. Secuencia	65
3.1.4.3. Integración	66
3.1.4.4. Articulación	67
3.1.5. Diseño de estrategias de enseñanza, aprendizaje y evaluación	67
3.1.5.1. Construcción activa	68
3.1.5.2. Aprendizaje situado	68
3.1.5.3. Interacciones sociales	69
3.1.5.4. Herramientas cognitivas	70
3.1.5.5. Estrategias de evaluación.....	71
3.2. Diseño y construcción de la plataforma tecnológica	71

3.2.1. Registro y configuración del curso	72
3.2.1.1. Desarrollo	73
3.2.2. Registro y configuración de los proyectos	73
3.2.2.1. Desarrollo	75
3.2.2.1.1. Registro de grupos de estudiantes	75
3.2.2.1.2. Registro de los equipos de trabajo	75
3.2.2.1.3. Asignación de los proyectos y roles	76
3.2.2.1.4. Registro de las historias de usuario	77
3.2.3. Seguimiento y control de los proyectos.....	77
3.2.3.1. Desarrollo	79
3.2.3.1.1. Plan de liberación	79
3.2.3.1.2. Planeación de la iteración.....	84
3.2.3.1.3. Entregables de la iteración.....	85
3.2.4. Monitorización y control del proyecto	86
3.2.5. Comunicación formal e informal.....	86
3.2.6. Evaluación de los proyectos	88
4. Resultados experimentales	91
4.1. Caso de estudio.....	91
4.1.1. Participantes	91
4.1.2. Instrumentos	91
4.1.3. Desarrollo del caso de estudio	96
4.1.3.1. Preparación del curso.....	96
4.1.3.2. Ejecución del curso teórico.....	96
4.1.3.3. Ejecución de los proyectos de GSD	99
4.1.3.3.1. Primera ronda de proyectos	108
4.1.3.3.2. Segunda ronda de proyectos	115
4.1.3.3.3. Tercera ronda de proyectos.....	122
4.1.4. Análisis de los resultados obtenidos.....	128
4.1.4.1. Evaluación del aprovechamiento académico.....	128
4.1.4.2. Evaluación del rendimiento de los estudiantes en los proyectos.....	130
4.1.4.3. Evaluación de la motivación de los estudiantes	135
4.1.4.4. Evaluación de la satisfacción de los estudiantes	136
4.1.4.5. Evaluación de la opinión de profesores.....	137
4.1.4.6. Consideraciones finales sobre los resultados obtenidos	138
4.2. Amenazas a la validez del caso de estudio	139
5. Conclusiones	141
6. Anexo A.- Acrónimos	145
7. Bibliografía.....	147

Lista de tablas

Tabla 1. Ejemplo para mostrar la diferencia entre las habilidades duras y las habilidades suaves. ...	4
Tabla 2. Habilidades suaves más valiosas que deben poseer los ingenieros de software (Maturro et al., 2015).....	4
Tabla 3. Problemas y retos del GSD y su relación con las habilidades (Holtkamp, Lau, y Pawlowski, 2015).	5
Tabla 4. Habilidades técnicas y generales para equipos de GSD (Saldaña-Ramos et al., 2014).	7
Tabla 5. Clasificación de trabajos analizados en la revisión sistemática de literatura (Clear et al., 2015).....	24
Tabla 6. Principales recomendaciones acerca de la enseñanza de GSD (Clear et al., 2015)	28
Tabla 7. Cuadro comparativo de investigaciones enfocadas en mejorar la enseñanza del GSD	50
Tabla 8. Modelos curriculares derivados del modelo de producto y del modelo de procesos (O'Neill, 2015).....	61
Tabla 9. Atributos por desarrollar en los estudiantes mediante el marco de referencia propuesto.	63
Tabla 10. Relación entre temas del marco de referencia y el conocimiento del estudiante.	67
Tabla 11. Cuestionario de evaluación de conocimientos del GSD.....	92
Tabla 12. Encuesta aplicada para obtener el grado de aceptación de Cadxela GSD por parte de los profesores.	95
Tabla 13. Proyectos pre-cargados en Cadxela GSD.....	99
Tabla 14. Casos de uso para el sistema de organización de un torneo de fútbol siete.	103
Tabla 15. Clases de complejidad de actores en el método de UCP.....	104
Tabla 16. UAW para el sistema de organización de un torneo de fútbol siete.....	105
Tabla 17. Clases de complejidad de casos de uso en el método de UCP.	105
Tabla 18. UUCW para el sistema de organización de un torneo de fútbol siete.	105
Tabla 19. TCF para el sistema de organización de un torneo de fútbol siete.	106
Tabla 20. EF para el sistema de organización de un torneo de fútbol siete.....	106
Tabla 21. Resultados de la evaluación de conocimientos.	129
Tabla 22. Resultados de la encuesta para evaluar la motivación de los estudiantes.	135
Tabla 23. Resultados de la encuesta para evaluar la satisfacción de los estudiantes.	136
Tabla 24. Resultados de la aplicación de la encuesta de opinión de profesores.....	137

Lista de figuras

Figura 1.1. Esquema del paradigma de GSD desde la perspectiva del marco de referencia propuesto	13
Figura 1.2. CWS enfocado a líder de proyecto (modificado de Palacio et al., 2011)	14
Figura 2.1. Estudios analizados por la investigación de educación de GSD (Monasor et al., 2010)	23
Figura 2.2. Distribución en áreas de interés de las recomendaciones del estudio de Clear et al. (2015)	28
Figura 2.3. Trabajos relacionados en la mejora o análisis de los procesos software (Monasor, 2014).....	30
Figura 2.4. Arquitectura simplificada de iBistro (diagrama de clases obtenido de Braun et al., 2002).....	38
Figura 2.5. Herramienta para estructurar la información de una reunión (traducido de Braun et al., 2002).....	39
Figura 2.6. Interfaz de estudiante para GSDGame (Valencia et al., 2016)	45
Figura 2.7. Interfaz para agregar problemas al juego (Valencia et al., 2016)	46
Figura 2.8. Interfaz para agregar un nuevo proyecto al juego (Valencia et al., 2016)	46
Figura 3.1. Metodología para el diseño curricular en la educación superior (O’Neill, 2015).....	54
Figura 3.2. Cobertura sobre la enseñanza del GSD en las universidades de México.....	58
Figura 3.3. Cobertura sobre la enseñanza del GSD en las universidades internacionales	59
Figura 3.4. Modelo de producto para el diseño curricular (O’Neill, 2015)	60
Figura 3.5. Modelo de procesos para el diseño curricular (O’Neill, 2015).....	61
Figura 3.6. Modelo de CWS que será implementado en la herramienta cognitiva (plataforma computacional)	70
Figura 3.7. Diagrama de flujo de ejecución de un curso de GSD a través de Caxela GSD	72
Figura 3.8. Diagrama de flujo para el registro y configuración de un curso de GSD	73
Figura 3.9. Interfaces para el registro y configuración de cursos.....	74
Figura 3.10. Registro de grupo de estudiantes.....	75
Figura 3.11. Registro de equipos de trabajo	76
Figura 3.12. Asignación de roles y proyectos	76
Figura 3.13. Registro de historias de usuario del proyecto	77
Figura 3.14. Proceso para el desarrollo de los proyectos	79

Figura 3.15. Proceso de planeación de la liberación (Cohn, 2005)	80
Figura 3.16. Pasos sugeridos por Cadxela GSD para la creación de plan de liberación	81
Figura 3.17. Reunión del equipo a través de Cadxela GSD	82
Figura 3.18. Estimación de historias de usuario a través de Cadxela GSD	83
Figura 3.19. Estimación de velocidad a través de Cadxela GSD	84
Figura 3.20. El profesor puede ver el progreso del equipo.....	84
Figura 3.21. Tablero de tareas en Cadxela GSD	86
Figura 3.22. Estadísticas del progreso del proyecto	86
Figura 3.23. Concepto de CWS en Cadxela GSD	87
Figura 3.24. Comunicación formal en Cadxela GSD	87
Figura 3.25. Información por estudiante en Cadxela GSD.....	88
Figura 3.26. Información de reuniones formales del equipo	89
Figura 4.1. Curso teórico acerca del GSD	97
Figura 4.2. Estudiantes identificaron a los roles involucrados en el ejercicio.....	98
Figura 4.3. Estudiantes aprendiendo a aplicar la técnica de estimación por póker	98
Figura 4.4. Estudiantes respondiendo el cuestionario de pre-evaluación	99
Figura 4.5. Proyecto asignado a un equipo en Cadxela GSD	108
Figura 4.6 Desempeño del equipo A en la primera ronda de proyectos.....	110
Figura 4.7 Desempeño del equipo B en la primera ronda de proyectos.....	111
Figura 4.8 Desempeño del equipo C en la primera ronda de proyectos.....	112
Figura 4.9 Desempeño del equipo D en la primera ronda de proyectos.....	113
Figura 4.10 Desempeño del equipo E en la primera ronda de proyectos.....	114
Figura 4.11 Desempeño del equipo A en la segunda ronda de proyectos.....	116
Figura 4.12 Desempeño del equipo B en la segunda ronda de proyectos.....	117
Figura 4.13 Desempeño del equipo C en la segunda ronda de proyectos.....	118
Figura 4.14 Desempeño del equipo D en la segunda ronda de proyectos.....	119
Figura 4.15 Desempeño del equipo E en la segunda ronda de proyectos.....	120
Figura 4.16. Ejemplo de una reunión donde participaron solamente dos estudiantes	121
Figura 4.17 Desempeño del equipo A en la tercera ronda de proyectos.....	123
Figura 4.18 Desempeño del equipo B en la tercera ronda de proyectos.....	124
Figura 4.19 Desempeño del equipo C en la tercera ronda de proyectos.....	125
Figura 4.20 Desempeño del equipo D en la tercera ronda de proyectos.....	126
Figura 4.21 Desempeño del equipo E en la tercera ronda de proyectos.....	127
Figura 4.22. Estudiantes respondiendo la pre-evaluación (imagen izquierda) y la post-evaluación (imagen derecha)	128
Figura 4.23. Ejemplo de proyecto incompleto entregado en la primera ronda de proyectos	131
Figura 4.24. Ejemplo de un problema de comunicación y trabajo en equipo suscitado en la primera ronda	132
Figura 4.25. Ejemplo de proyecto entregado en la segunda ronda de proyectos.....	132

Figura 4.26. Tablero de reuniones conducidas por un equipo durante la segunda ronda.....	133
Figura 4.27. Ejemplo de proyecto entregado en la tercera ronda.....	134
Figura 4.28. Ejemplo de una reunión exitosa en la tercera ronda de proyectos	134

Resumen

Hoy en día es común que las empresas de desarrollo de software utilicen la subcontratación de personal para abaratar los costos de desarrollo, contratando programadores en países donde el salario mínimo es menor, pero arriesgando posiblemente, y como consecuencia, la calidad del producto final. Esta forma de desarrollar el software ha sido denominada “Desarrollo Global de Software” (GSD, por sus siglas en inglés) y ha introducido beneficios importantes a la industria actual, principalmente el abaratamiento de costos de desarrollo, la unión de diferentes ideas y culturas, entre otros. No obstante, el paradigma del GSD se encuentra en constante evolución, por lo que aún se reportan diversos problemas y retos relacionados con las distancias geográfica, temporal y cultural a las que se enfrenta este enfoque.

En este contexto, la enseñanza del GSD en las universidades es considerada una alternativa importante para alcanzar sus beneficios. Por tal motivo, en esta tesis se presenta un marco de referencia que establece los lineamientos técnico-metodológicos para llevar a cabo un curso de GSD a nivel licenciatura, que permita a los estudiantes obtener experiencia práctica acerca del enfoque y, al mismo tiempo, les ayude a mejorar sus habilidades de comunicación y trabajo en equipo. Además, se presenta una plataforma computacional llamada Cadxela GSD (o Descubre GSD, en su traducción de Zapoteco a Español) que permite llevar a la práctica los lineamientos del marco de referencia.

La plataforma y el marco de referencia fueron evaluados a través de la ejecución de un caso de estudio que contó con la participación de 20 estudiantes y 5 profesores de 5 universidades distintas, quienes llevaron a cabo un curso de GSD, en donde se impartieron los conceptos teóricos del enfoque y después se desarrollaron tres rondas de proyectos prácticos. Los resultados obtenidos demostraron que, a través de la ejecución práctica de los proyectos, los estudiantes mejoraron sus habilidades de comunicación y trabajo en equipo. Así mismo, los estudiantes mejoraron su conocimiento acerca del GSD y se sintieron motivados y satisfechos con su desempeño. Finalmente, la plataforma fue considerada, por los profesores participantes, como una herramienta útil en la enseñanza del GSD.

Abstract

It is common for software development companies to use the outsourcing of staff to reduce the costs of developing and hiring programmers in countries where there are lower labor costs, but which also risks compromising the quality of the final product. This form of software development has been called "Global Software Development" (GSD) and it has introduced important benefits to the software industry mainly by reducing costs, the bringing together of different ideas and cultures, among others. However, the GSD paradigm is constantly growing, and various problems and challenges related to geographical, temporal and cultural distances are still being reported.

In this context, the teaching of GSD in universities is considered an important alternative in achieving GSD benefits. For this reason, in this thesis a framework that establishes the methodological-technician guidelines for creating an undergraduate GSD course is presented that allows students to acquire practical experiences with the GSD approach and, at the same time, improve communication and teamwork skills. Additionally, a computational platform is presented called Caxela GSD (which means 'Learn GSD', when translated from Zapotec to Spanish) that allows the guidelines of the reference framework to be put into practice.

The platform and the framework were evaluated by implementing a case study that included the participation of 20 students and 5 professors from 5 different universities, who delivered a GSD course, in which, the theoretical concepts of the approach were taught and three iterations of practical projects were developed. The results obtained showed that, through the practical execution of the projects, the students improved their communication and teamwork skills. Similarly, the students improved their knowledge on GSD and were motivated and satisfied with their performance. Finally, the platform was considered as a useful tool in the teaching of the GSD by the participating.

1. Introducción y motivaciones

1.1. Contexto del problema

El desarrollo tecnológico ha permitido que diversas industrias extiendan su alcance y su participación en los mercados mundiales, ya que el creciente avance tecnológico en tópicos relacionados con las telecomunicaciones, el transporte, y la tecnología en sí, ha permitido que diversas industrias crezcan de tal manera que hoy en día sea común que una empresa pueda ubicarse y sea productiva en diferentes partes del mundo al mismo tiempo. En este contexto, las empresas de desarrollo de software y tecnología también se han unido a este movimiento dando lugar así a lo que se denomina Desarrollo Global de Software (GSD, por sus siglas en inglés). De acuerdo con Vizcaíno, García, Piattini, y Beecham (2016), el concepto de GSD surgió hace más de 20 años cuando la industria de software comenzó a practicar la subcontratación, pero no se hizo oficial hasta el 2006 con la celebración de la Primera Conferencia Internacional sobre Desarrollo Global (*1st IEEE International Conference on Global Software Engineering*) realizada en Brasil. A partir de entonces, el GSD se ha consolidado como una de las prácticas más relevantes en la investigación y uso de la Ingeniería de Software (IS).

El GSD permite a las empresas abaratar los costos de desarrollo al mismo tiempo que intenta mantener el nivel de calidad del producto, esto debido principalmente a que se forman equipos de trabajo multidisciplinarios en donde participan personas de diferentes lugares que a menudo experimentan los siguientes beneficios:

- Trabajar con profesionales de todo el mundo sin tener que afrontar el costo de traslado de las personas. De esta manera se pueden contratar desarrolladores de software en países donde los salarios son más reducidos mediante la subcontratación de una empresa o la creación de sucursales de la misma empresa en otros países (Damian y Moitra, 2006).
- Producir software para clientes remotos sin necesidad de trasladar el equipo de desarrolladores, incrementando de esta forma las posibilidades de introducirse en nuevos mercados (Damian y Moitra, 2006; Colomo-Palacios, Casado-Lumbreras, Soto-Acosta, García-Peñalvo, y Tovar, 2014).
- Realizar jornadas de trabajo más extensas cuando los desarrolladores se encuentran ubicados en sitios con diferente zona horaria, lo cual permite que la empresa alcance una mayor productividad (Carmel y Agarwal, 2001; Ebert y De Neve, 2001; Herbsleb y Moitra, 2001).

- Obtener mayor experiencia, conocimiento técnico y destrezas que son compartidas entre los diferentes participantes del equipo de trabajo y los *stakeholders*¹ distribuidos en diferentes partes (Ebert y De Neve, 2001; Colomo-Palacios et al., 2014).

De acuerdo con Vizcaíno, García, y Piattini (2015), existen otros factores que hacen interesante este paradigma de desarrollo, tal es el caso de la innovación que se produce de la diversidad cultural y de compartir experiencias entre los integrantes del equipo. Además, los equipos que trabajan bajo el esquema de GSD deben colaborar en proyectos geográficamente distribuidos y, por lo tanto, incluyen múltiples participantes de diferentes países, culturas, y lenguas. Por otro lado, Saldaña-Ramos, Sanz-Esteban, García, y Amescua (2014) mencionan que las principales diferencias entre los equipos de trabajo de GSD y los equipos tradicionales son las siguientes:

- Los participantes de un equipo de GSD realizan tareas interdependientes a través de diferentes ubicaciones geográficas, lo que origina que el conocimiento sobre cada tarea sea usualmente distribuido en diferentes sitios.
- Los miembros de un equipo de GSD suelen tener distintas lenguas maternas y culturas, lo que suele dificultar la comunicación, además ésta depende en gran medida de la tecnología de cada sitio.
- La dinámica de trabajo en un equipo de GSD debe ajustarse debido a la diferencia de horarios, lo cual dificulta la gestión de la agenda de trabajo.

Por otro lado, el GSD también produce un alto impacto en la manera en que se conciben, diseñan, construyen, prueban, y entregan los productos de software a los clientes (Herbsleb y Moitra, 2001), de tal manera que los métodos de trabajo para equipos tradicionales no puedan aplicarse correctamente a equipos de GSD, por lo que es necesario desarrollar nuevos métodos de trabajo (Jiménez, Piattini, Vizcaino, 2009). Aunado a esto, Conchúir (2010) afirmó que el GSD introdujo una serie de problemas que se pueden agrupar en las denominadas “3 C’s”: desafíos en la Comunicación (e.g., el intercambio de conocimientos e información entre los integrantes del equipo), desafíos en la Coordinación (e.g., realización de tareas), y desafíos en el Control (e.g., gestión del proyecto). Estos desafíos se acentúan debido a lo que Ågerfalk et al. (2005) denominaron como las tres distancias:

- Distancia geográfica, definida como: “la medida de esfuerzo que un individuo necesita realizar para visitar otro punto, alejado del primero”.
- Distancia temporal, definida como: “la medida de la deslocalización en tiempo, experimentada por dos individuos que desean interactuar”.
- Distancia sociocultural definida como: “la medida en que un individuo comprende las costumbres (símbolos, normas, y valores sociales) y cultura de otro individuo”.

¹ El término “*stakeholder*” agrupa a trabajadores, organizaciones, accionistas y proveedores, entre muchos otros actores clave que se ven afectados por el producto de software que se pretende desarrollar. Generar confianza en ellos es fundamental para el éxito en el desarrollo del software, dado que son éstos los que proporcionarán las necesidades que deberán ser cubiertas (Pacheco y García, 2012).

Por lo tanto, el GSD presenta aún una serie de problemas a resolver, motivo por el cual se ha convertido en un tópico importante de investigación dentro de la IS. Con respecto a lo anterior, Saldaña-Ramos et al. (2014) afirman que diversas organizaciones han implementado el GSD y comprobado que formar equipos bajo este enfoque resulta complejo, principalmente por los cuatro desafíos que a menudo enfrentan:

- Falta de entendimiento de los objetivos del proyecto, lo cual provoca que los integrantes del equipo se sientan aislados y se reúsen a colaborar y compartir conocimientos.
- Dificultades en la comunicación, debido principalmente a las distancias geográficas y a las diferencias horarias.
- Las diferencias entre los mecanismos de gestión individual y las habilidades de cada integrante del equipo, lo cual provoca cuellos de botella en la ejecución del proyecto.
- Ineficiencia en la gestión del conocimiento compartido entre los miembros del equipo, lo cual provoca inconsistencias por la falta de conocimiento de los activos del proyecto.

Es importante destacar que las habilidades del personal que trabaja bajo el esquema del GSD juegan un rol importante para alcanzar el éxito del proyecto. Por tal motivo, las competencias son importantes para mejorar la calidad del capital humano de las organizaciones de software. Es decir, si el personal está bien capacitado, hará sus tareas de manera más eficiente. De acuerdo con Sagi-Vela (2004), una competencia se define como: *“los conocimientos, habilidades, y actitudes que se aplican para llevar a cabo cierta responsabilidad o contribución profesional que ayude a asegurar el éxito”*. En este sentido, el nivel de competencia del personal de una organización de software es el factor más importante para alcanzar el éxito (Saldaña-Ramos et al., 2014). En el contexto del desarrollo de software, Ahmed, Capretz, Bouktif, y Campbell (2012) mencionan que el software es producto de las actividades humanas, que a menudo involucran capacidades de resolución de problemas, aspectos cognitivos, e interacción social. Así mismo, cada fase del desarrollo de software (e.g., análisis de requisitos, diseño, construcción, pruebas, y mantenimiento) requiere que los individuos involucrados posean un conjunto de habilidades que no necesariamente serán comunes para todas las actividades del desarrollo. Con relación a esto, las habilidades requeridas por la industria de software se dividen en dos grandes categorías: habilidades duras (i.e., conocimientos y habilidades técnicas que un individuo debe tener para llevar a cabo las tareas asociadas con su empleo) y habilidades suaves² (i.e., rasgos de la personalidad, habilidades de la interacción social, comunicación, y hábitos personales). De modo ilustrativo, la Tabla 1 muestra un ejemplo de una oferta de trabajo para un ingeniero de software, en donde se definen claramente las diferencias entre estos tipos de habilidades.

En este contexto, Maturro, Raschetti, y Fontán (2015) realizaron un estudio en Uruguay con 11 empresas de desarrollo de software con el objetivo de determinar las habilidades suaves más valiosas que un ingeniero de software debe poseer. Dicho estudio, que involucró a 18 jefes de proyectos y 17 miembros de equipos de desarrollo de software, evidenció que el análisis, la resolución de problemas, el compromiso, la responsabilidad, el trabajo en equipo, el afán de

² La investigación de Ardis, Chenoweth, y Young (2008) ha clasificado como “suaves” las diferentes habilidades sociales que los profesionales de la Ingeniería de Software deben desarrollar. Concretamente la observación, la revisión, la presentación oral de la información, la escritura, la planificación, la cooperación, la negociación, la reflexión, el liderazgo, el trabajo en equipo, y el juicio son habilidades sociales que dichos profesionales deberían adquirir durante su formación.

aprender, y las habilidades de comunicación son consideradas habilidades valiosas para un ingeniero de software.

Tabla 1. Ejemplo para mostrar la diferencia entre las habilidades duras y las habilidades suaves.

Habilidades duras	Habilidades suaves
Experiencia en la programación de SQL, Java, JavaScript, aplicaciones <i>web</i> y bases de datos relacionales (SQL Server, Oracle), experiencia en la arquitectura J2EE.	Habilidades analíticas fuertes. Excelente habilidad de comunicación verbal y escrita. Fuertes habilidades interpersonales y habilidad de resolución de problemas. Habilidad para trabajar de manera independiente y habilidad de aprendizaje rápido.

Los resultados obtenidos en este estudio se resumen en Tabla 2 la cual muestra las 15 habilidades suaves consideradas como las más valiosas, de acuerdo con las respuestas de los 35 participantes.

Tabla 2. Habilidades suaves más valiosas que deben poseer los ingenieros de software (Matturro et al., 2015).

Habilidades suaves	%
Análisis, resolución de problemas	91.4
Compromiso, responsabilidad	88.6
Trabajo en equipo	88.6
Afán de aprender	77.1
Habilidades de comunicación	74.3
Iniciativa, proactivo	68.6
Motivación	68.6
Orientado a resultados	68.6
Autonomía	65.7
Habilidades interpersonales	48.6
Habilidades organizativas	48.6
Orientación al cliente	48.6
Habilidades de planeación	45.7
Metódico	45.7
Liderazgo	42.9

Por lo tanto, es importante destacar que las competencias individuales constituyen un factor de éxito en el desarrollo de software y la comunicación, la motivación, y el trabajo en equipo son competencias esenciales que cada profesional en el área debería poseer. De la misma manera que en el desarrollo de software tradicional, las habilidades suaves son requeridas para los equipos de GSD; sin embargo, existen algunas habilidades que estos equipos deben desarrollar de manera específica, como es el caso de la comunicación y el trabajo en equipo (Saldaña-Ramos et al., 2014). Así mismo, el GSD introduce varios desafíos y problemas para los miembros del equipo. Cuando se enfrentan a un cambio contextual en su entorno de trabajo, los individuos tienen que adaptarse a la nueva situación. Esto incluye la adaptación de estilos de trabajo, comportamientos y métodos.

Igualmente, se deben abordar nuevos desafíos, especialmente aquellos basados en el trabajo virtual y el trasfondo cultural de los miembros del equipo (Holtkamp, Lau, y Pawlowski, 2015).

1.2. Importancia del problema

Los participantes de los equipos de GSD deben poseer ciertas habilidades que les permitan involucrarse en los procesos de desarrollo globales. Bhatti y Ahsan (2016), por ejemplo, afirman que es fundamental examinar a las personas que participan en el proceso, sus habilidades individuales y colectivas, y el comportamiento que deben mostrar (individual e interactivo) para implementar el proceso. Así pues, el éxito de los proyectos de GSD radica en gran medida en el personal que trabaja bajo este paradigma y por tal motivo es importante que éste posea ciertas habilidades que le permita desenvolverse adecuadamente en este entorno. En este sentido, la Tabla 3 muestra los principales problemas y retos del GSD y su relación con la falta de habilidades de los ingenieros de software que trabajan bajo este esquema.

Tabla 3. Problemas y retos del GSD y su relación con las habilidades (Holtkamp, Lau, y Pawlowski, 2015).

Problema/Reto	Descripción	Proceso	Categoría de la habilidad
Problemas organizativos	Las diferencias en las leyes, las tradiciones, y las reglamentaciones requieren un esfuerzo adicional para hacer cumplir las normas y procesos laborales	Gestión de proyectos Desarrollo de la gestión de procesos	Gestión de proyectos Coordinación
Diferentes entendimientos conceptuales	Los conceptos y términos son entendidos de diferente manera debido a la diversidad cultural	Formación de equipos Asignación de proyectos Desarrollo	Cultura
Diferentes herramientas o uso de las herramientas	La selección de herramientas depende en gran medida de los factores organizacionales y culturales	Desarrollo	Herramientas/ Tecnologías de la Información
Colaboración asíncrona	Sincronización del trabajo entre diferentes zonas geográficas y diferentes zonas horarias	Desarrollo	Coordinación Colaboración Comunicación
Falta de relaciones interpersonales	Los lazos débiles y el grado de familiaridad entre los miembros del equipo pueden causar varios problemas, como la falta de conciencia del equipo y la incertidumbre sobre los contactos correctos	Desarrollo	Cultura
Falta de confianza	La incertidumbre sobre los comportamientos de trabajo y las competencias de los miembros del equipo conducen a una falta de confianza	Desarrollo	Cultura

Problema/Reto	Descripción	Proceso	Categoría de la habilidad
Problemas de comunicación	La comunicación requiere un esfuerzo adicional para evitar la falta de comunicación informal, pérdida de la riqueza de la comunicación y malentendidos	Desarrollo	Comunicación
Lenguaje	Las diferencias de idioma pueden causar malentendidos, retrasos y errores	Desarrollo	Comunicación
Diferentes antecedentes	Los diferentes antecedentes de los miembros del equipo conducen a diferentes puntos de vista y malentendidos	Formación de equipos Asignación de proyectos Desarrollo	Cultura
Diferencias en las negociaciones y aceptación del trabajo	Los antecedentes culturales pueden influir en el resultado percibido de las negociaciones y distribuciones de tareas	Formación de equipos Desarrollo	Cultura
Valores, normas, y prácticas	El trasfondo cultural tiene un fuerte impacto en los valores, normas y prácticas individuales	Desarrollo	Cultura
Percepción del tiempo y comportamiento basado en el tiempo	Los miembros del equipo de diferentes orígenes pueden tener una percepción diferente del tiempo y plazos de entrega	Desarrollo Gestión de proyecto	Cultura

Aunado a lo anterior, Paasivaara, Lassenius, Damian, Rätty, y Schröter (2013), mencionan que los desafíos más importantes de GSD, tales como las diferencias culturales, diferencias de lenguaje, los problemas de las jornadas de trabajo, la falta de contacto frecuente, la falta de confianza entre los integrantes del equipo y la falta de confianza para comunicarse abiertamente, están estrechamente relacionados con la comunicación. A su vez, Casey y Richardson (2008) sugieren que los ingenieros de software que trabajan bajo el paradigma de GSD posean las siguientes habilidades:

- Resolución de conflictos, que incluye entre otras la capacidad para hacer frente a situaciones difíciles y conflictivas; el diagnóstico temprano de conflictos en el equipo virtual; y la actitud positiva y la capacidad de motivación.
- Trabajo en equipo, considerando la capacidad para pensar desde la perspectiva del interlocutor o la habilidad para ganar la confianza del equipo, el uso de estructuras de gratificación y recompensa o respuesta apropiada ante críticas o sugerencias.
- Destrezas comunicativas, como el dominio de la lengua común empleada por la organización o el conocimiento de protocolos de comunicación y costumbres de las diferentes culturas, entre otras.

De manera similar, Vardi (2010), apoyado por la Asociación para la Maquinaria Informática (ACM, por sus siglas en inglés), determinó que los ingenieros de software debían estar mejor preparados y menciona específicamente la necesidad de la capacitación en habilidades suaves, tales como el trabajo en equipo, la comunicación, la organización de los procesos, y temas relacionados con el desarrollo global. Igualmente, la investigación de Saldaña-Ramos et al. (2014), identifica un conjunto de habilidades técnicas y generales que los integrantes de los equipos de GSD deben poseer (véase Tabla 4).

Tabla 4. Habilidades técnicas y generales para equipos de GSD (Saldaña-Ramos et al., 2014).

Habilidades técnicas	Habilidades generales
<ul style="list-style-type: none"> • Identificación de competencias • Gestión de la comunicación síncrona y asíncrona en contextos globales • Identificación de requisitos • Estimación y priorización de necesidades • Gestión de requisitos • Recopilar, analizar e interpretar la información • Solución de problemas técnicos • Gestión del conocimiento compartido • Gestión de las sesiones de lluvia de ideas • Análisis de sinergias a partir de la información obtenida de redes sociales • Uso de técnicas avanzadas para la comunicación distribuida 	<ul style="list-style-type: none"> • Resolución de conflictos interpersonales en entornos distribuidos • Actitud positiva y capacidad de motivación • Capacidad de autoaprendizaje • Habilidad para las relaciones interpersonales • Capacidad de utilizar las tecnologías de la comunicación y la información • Habilidad para trabajar en un entorno global • Capacidad para comunicarse oralmente y por escrito en inglés • Capacidad de organización y planificación • Iniciativa y liderazgo • Toma de decisiones

En este contexto, una de las principales estrategias que permitiría minimizar los problemas en el GSD, considerando que el personal cuente o no con tales habilidades suaves, consiste en proporcionar una formación adecuada a los futuros ingenieros de software. Dicha formación debe permitir que el estudiante adquiera conocimientos, aptitudes, habilidades y destrezas, es decir, competencias que le permitan desenvolverse de manera efectiva en entornos de desarrollo global (Vizcaíno et al., 2015). Sin embargo, los cursos tradicionales de IS ofrecidos por las universidades no cubren completamente los nuevos desafíos que el GSD plantea. Al respecto, Paasivaara et al. (2013) mencionan que el número de experiencias reportadas sobre la enseñanza del GSD ha ido creciendo desde principios del siglo XXI. Estas investigaciones hacen hincapié en que la enseñanza de este paradigma debe apoyarse en experiencias prácticas a través de las cuales los estudiantes puedan aprender haciendo (*learning-by-doing*). El estudio de Fortaleza, Conte, Marczak, y Prikladnicki (2012), por ejemplo, reveló que de 19 cursos sobre GSD solamente uno era impartido de manera teórica, mientras que todos los demás tenían un proyecto distribuido como elemento central. Así pues, la mayoría de los cursos de GSD debían fortalecer las habilidades suaves que permitan a los estudiantes afrontar los desafíos que este paradigma plantea. Los principales desafíos reportados en este estudio incluían a los problemas de comunicación, dificultades de lenguaje, asuntos sociales e interculturales, y temas motivacionales. Por otro lado, la literatura pedagógica sobre el GSD ha identificado las habilidades que los profesionales necesitan y, por lo tanto, que son importantes en la formación de los estudiantes, tales como: comunicación regular con los miembros

distribuidos del equipo, dinámicas de equipo trabajando en equipos culturalmente diferentes, manejo del tiempo y uso de tecnologías colaborativas (Paasivaara et al., 2013). Del mismo modo, Monasor, Vizcaíno, y Piattini (2011) identificaron las habilidades que los estudiantes deberían desarrollar durante un curso sobre el GSD:

- Entendimiento sobre protocolos de comunicación y costumbres de las diferentes culturas.
- Capacidad para comunicarse eficientemente usando una terminología y lengua comunes.
- Habilidad para ganarse la confianza del equipo y para resolver conflictos.
- Conocimiento de técnicas de negociación y redacción de contratos.
- Gestión de la ambigüedad e incertidumbre, derivada de los problemas de comunicación.
- Uso de herramientas de gestión del conocimiento, gestión de documentos, y de control de versiones.
- Capacidad para liderar las reuniones y gestionar el tiempo.
- Habilidad para trabajar en equipo y pensar desde la perspectiva del interlocutor.
- Capacidad de improvisación y habilidad para la comunicación informal.
- Habilidad para tratar con un equipo multidisciplinario.

Considerando lo anterior, el estudio de Monasor et al., (2011) menciona que algunas universidades organizan sus cursos de GSD con segmentos de trabajo práctico que involucra la participación de organizaciones asociadas (regularmente otras instituciones educativas nacionales o internacionales). Esta estrategia tiene como objetivo preparar a los estudiantes de la manera más realista posible para los desafíos que enfrentarán cuando trabajen en proyectos reales en entornos de GSD. No obstante, la organización de este tipo de cursos colaborativos suele implicar diversas actividades para los profesores que van desde la adaptación del método de enseñanza y los horarios de los cursos, hasta la organización de posibles infraestructuras de comunicación con el objetivo de que los conocimientos previos de los estudiantes se correspondan estrechamente entre sí (Lappalainen, Tripathi, y Similä, 2016). En consecuencia, organizar este tipo de cursos en colaboración con otras universidades no es una tarea sencilla, ya que de acuerdo con Matthes et al. (2011), las instituciones que pretenden implementar estos cursos deben realizar un esfuerzo adicional para organizarlos y coordinarlos. Además, los diferentes horarios semestrales en combinación con la desigualdad de los sistemas de calificación y evaluación complican la cooperación entre las universidades distribuidas alrededor del mundo. Otro punto por considerar es que la enseñanza del GSD está limitada por el periodo de formación, que suele ser mediante un curso de duración de un semestre (cinco meses) o menos. Sin embargo, a menudo se necesita más tiempo y esfuerzo para lograr una reproducción exitosa de un entorno de GSD en contextos educativos y que, como consecuencia, se obtenga un nivel adecuado de aprovechamiento (Crnković, Bosnić, y Žagar, 2012).

1.3. Necesidad de solución

Los cursos de GSD a menudo se imparten en los últimos años de estudio, justo antes de que los estudiantes tengan la oportunidad de incorporarse al mercado laboral. De esta manera, los egresados tienen la posibilidad de aplicar las habilidades técnicas que aprendieron durante sus

estudios en un entorno cercano a un proyecto real, así como adquirir habilidades suaves que son cruciales para el éxito de los proyectos de GSD y de cualquier tipo de desarrollo de software. Ahora bien, la mayoría de los estudios sobre la enseñanza del GSD mencionan problemas organizativos cuando se trata de lograr la colaboración entre estudiantes localizados en diferentes ubicaciones geográficas. Por otra parte, los estudiantes que participan en estas actividades poseen diferentes niveles de conocimiento o habilidades, lo que hace necesario capacitar a cada estudiante de manera diferente. Así mismo, la cultura y lenguaje nativo de cada participante debe igualmente tenerse en cuenta en el diseño de dicha capacitación. Otro de los problemas destacados es la ineficacia de la comunicación a través de medios comunes como el correo electrónico o el *chat*, en ocasiones relacionados con cuestiones técnicas, lo que generalmente conlleva al incumplimiento de los plazos fijados al inicio del proyecto (Vizcaíno et al., 2015).

Durante los últimos años se han realizado diversas investigaciones con el objetivo de abordar los problemas relacionados con la enseñanza del GSD en entornos universitarios. La investigación de Ende, Lämmermann, Brockmann, y Ayurzana (2013), por ejemplo, se enfocó a aplicar el enfoque de Aprendizaje basado en Problemas (PBL, por sus siglas en inglés) en la mejora de la enseñanza del GSD a través de un proyecto desarrollado por dos equipos de estudiantes de dos diferentes países: Mongolia y Alemania. En éste participaron cuatro profesores de la Universidad de Ciencia y Tecnología de Mongolia (MUST, por sus siglas en inglés) quienes realizaron el rol de “clientes internacionales”, mientras que los estudiantes del Instituto Tecnológico de Nuremberg (NIT, por sus siglas en inglés), además de obtener los requisitos de software relacionados con el proyecto, utilizaron una metodología de desarrollo ágil (i.e., *Scrum*), por lo que debieron estar en constante comunicación. El objetivo del proyecto fue que los estudiantes adquirieran las habilidades necesarias para trabajar en proyectos de GSD. Este curso lo han implementado estas universidades desde el año 2012 y se han integrado equipos distribuidos en diferentes zonas geográficas para participar en el desarrollo de proyectos. Los hallazgos obtenidos evidenciaron desafíos lingüísticos y culturales durante las comunicaciones virtuales, problemas técnicos con las conexiones de internet, las diferencias de zonas horarias, y las distancias geográficas.

Noll et. al. (2014) desarrollaron una propuesta un tanto diferente, un juego serio llamado GSD Sim. Dicho juego fue concebido para que los estudiantes adquirieran experiencia para solventar las dificultades que implica el GSD, desde el punto de vista de un líder de proyectos. El juego coloca al estudiante en el rol de líder de un proyecto de desarrollo de software que debe controlar a equipos de programadores ubicados en diferentes lugares alrededor del mundo, y asignarles el desarrollo de módulos que componen al producto de software en sí. En este escenario, el estudiante dispone de un presupuesto limitado para contratar a los programadores, por lo que si los costos de desarrollo exceden dicho presupuesto se aplica una penalización del 25% sobre el límite presupuestario. De manera similar, si el proyecto fuera entregado fuera del tiempo establecido, el total de ventas se reduciría cada mes después de la fecha límite. Por el contrario, si el proyecto se termina con anticipación, se añadirán meses de ingresos adicionales. De acuerdo con los autores, se pretende mostrar a los estudiantes la importancia de maximizar los beneficios, reduciendo los costos o aumentando los ingresos. Sin embargo, este juego no ha sido validado aún a pesar de que los autores mencionaron que se trabajaría con la premisa de probarlo en un entorno real.

La investigación de Lappalainen et al. (2016), planteó el desarrollo de un curso sobre GSD a nivel maestría para estudiantes de IS, con el objetivo de estudiar los desafíos de este enfoque y sus posibles soluciones. Dicho curso se llevó a cabo siguiendo dos enfoques: el primero se basó en clases teóricas para enseñarle a los estudiantes los fundamentos y los problemas relacionados con el desarrollo global, mientras que el otro enfoque consistió en realizar un ejercicio de simulación en un

proyecto real de GSD. Por un lado, las clases se centraron en los problemas de comunicación, las diferencias temporales, y las preocupaciones multiculturales. Adicionalmente, se analizaron los aspectos actuales de la investigación sobre el GSD, incluyendo casos de estudio relacionados con la industria. Por otro lado, el ejercicio de simulación sirvió para que un equipo virtual demostrara la aplicación del GSD utilizando el apoyo de métodos y herramientas apropiadas. En este sentido, los estudiantes formaron equipos de 3 a 4 integrantes y desarrollaron ejercicios virtuales durante cinco semanas con el objetivo de analizar los problemas que se presentan en un desarrollo global. Después del curso los estudiantes realizaron un informe de reflexión en donde analizaron su aprendizaje y las posibilidades que tendrían de aplicar el conocimiento adquirido en entornos reales. Los principales resultados, obtenidos a través de encuestas, evidenciaron que más del 50% de los estudiantes estuvo de acuerdo en que las clases se llevaron a cabo de una manera que aseguró su preparación y participación; alrededor del 95% de los estudiantes estuvo de acuerdo en que las clases proporcionaron un medio para elaborar y desarrollar su propia comprensión sobre el GSD; y el 82% de los estudiantes afirmó que la simulación les ayudó a reconocer los diversos aspectos que involucra el GSD.

A su vez, Li, Krusche, Lescher, y Bruegge (2016) realizaron la simulación de un proyecto global de software en un curso de IS en el que participaron 100 estudiantes. La simulación consistió en que los estudiantes debían desarrollar un proyecto considerando que se encontrarían ubicados en tres sitios diferentes: en un sitio (Brasil) se debía encontrar el equipo de desarrollo de la empresa sede; en un segundo sitio (Alemania) se encontraría un equipo de desarrollo subcontratado que debía colaborar con otro equipo localizado en Italia; y en el tercer sitio se ubicaría un grupo de personas con un conjunto único de tecnologías y conocimientos. Los estudiantes debían desempeñar estos roles considerando que el objetivo del proyecto era integrar la tecnología *MyVoice*, adquirida por la empresa de Brasil y proporcionada por el grupo de personas del tercer sitio, con un producto existente desarrollado en los otros dos sitios. El desarrollo del proyecto estableció algunas restricciones para los estudiantes: los miembros de cada equipo no deberían abandonar su sitio de trabajo, cada proyecto podía seleccionar los métodos de comunicación que se consideraran apropiados, las reuniones y entregables se tendrían que documentar y cada proyecto podía solicitar horas extra de trabajo, pero esto implicaría costos extra. Teniendo este objetivo en mente, se definieron cuatro proyectos para que los estudiantes integraran los grupos de trabajo correspondientes a los tres sitios definidos por el problema, escogieran sus medios de comunicación, y coordinaran las horas de trabajo dadas las diferencias horarias entre cada país. Las impresiones de los estudiantes fueron recogidas a través de encuestas que evidenciaron que el 50% de los estudiantes consideró que es difícil coordinar el tiempo de trabajo; más del 50% consideró que es posible que existiera una sobrecarga de comunicación; el 43% argumentó la carencia de comunicación informal; el 45% consideró que es difícil comunicar los cambios en los requisitos; y el 46% afirmó que es difícil compartir la experiencia entre los *stakeholders*.

En resumen, los trabajos anteriores muestran una clara tendencia de los investigadores por generar propuestas que mejoren la formación de los estudiantes en el GSD y, principalmente, que fortalezcan las habilidades que requieren para afrontar los problemas que este tipo de desarrollo trae consigo. Se han diseñado cursos que implementan la simulación como medio de enseñanza, también se han desarrollado juegos educativos y plataformas integrales que incluyen infraestructura, como es el caso de una fábrica de software. Pero, estas herramientas requieren la cooperación entre universidades, que en muchos casos resulta compleja por las diferentes formas de enseñar, los planes de estudio, los métodos de calificación, entre otros factores. Igualmente, es importante ofrecer alternativas que permitan que los estudiantes desarrollen sus habilidades suaves, en relación con proyectos de GSD, en un salón de clases. Con esta idea en mente esta tesis propone la creación

de un marco de referencia³ que establezca las pautas para llevar a cabo un curso de GSD entre dos o más universidades. De este modo, de manera adicional se diseñará una plataforma tecnológica para llevar a cabo dicho curso, cuyo objetivo principal será fomentar entre los estudiantes las habilidades suaves requeridas por los equipos de GSD. Para ello se establecerán ciertos lineamientos que permitan a los profesores organizar el curso, establecer los equipos de trabajo, establecer los métodos y formas de evaluación, y dar seguimiento al aprendizaje de los estudiantes a través de la plataforma creada.

1.4. Delimitaciones y limitaciones de la investigación

Esta tesis está delimitada y limitada por los siguientes puntos:

- El marco de referencia se enfocará a proporcionar las pautas y lineamientos para el desarrollo de un curso sobre el GSD entre dos o más universidades, no se considera la implementación de esta propuesta en una sola universidad.
- La propuesta se enfocará en mejorar específicamente dos habilidades suaves: la comunicación y el trabajo en equipo desde el contexto de un líder de proyectos.
- La propuesta estará enfocada a estudiantes de nivel licenciatura que cursan los últimos años de su formación universitaria.
- Dicha propuesta será evaluada mediante la implantación de una plataforma tecnológica como principal medio de soporte de un curso tradicional.
- El desarrollo de esta plataforma contemplará la infraestructura y disponibilidad del equipo informático en las universidades (e.g., computadoras, medios de comunicación, ancho de banda de internet, aulas equipadas).
- La tesis plantea la idea de tener un proyecto como elemento central del curso sobre el GSD. En este sentido, la plataforma tecnológica se enfocará en promover las habilidades suaves relacionadas con la gestión de un proyecto de GSD. Por lo tanto, se asume que el resto de las fases del ciclo de vida del software deberán ser impartidas por el profesor considerando el paradigma de GSD o, en su defecto, el paradigma tradicional.

1.5. Hipótesis de la tesis

La hipótesis de este trabajo se plantea como una relación causal y se enuncia de la siguiente manera:

“La creación e implementación de un marco de referencia permitirá que los estudiantes desarrollen las habilidades suaves de comunicación y trabajo en equipo en entornos de GSD”

1.6. Objetivos de la tesis

Los objetivos del trabajo están compuestos de un objetivo general y varios objetivos secundarios, los cuales se describen a continuación:

³ De acuerdo con el Comité para la Evaluación de Programas de Pedagogía y Educación, A. C. (CEPPE), dentro de un contexto educativo, un marco de referencia debe establecer los lineamientos técnico-metodológicos para realizar el proceso de enseñanza dentro de un programa educativo. Este marco debe establecer los pasos necesarios para la acreditación de cierta materia, los atributos a evaluar a través de los indicadores (tanto teóricos como prácticos), y la definición de cada una de las etapas del proceso de enseñanza-aprendizaje.

1.6.1. Objetivo general

Diseñar e implementar un marco de referencia que permita a dos universidades llevar a cabo un curso bajo el paradigma GSD, para mejorar las habilidades suaves de trabajo en equipo y comunicación de sus estudiantes al desempeñar el rol de líder de proyecto.

Para alcanzar este objetivo general será necesario conseguir ciertos objetivos secundarios. Estos establecerán las aportaciones esperadas al final de la tesis:

1.6.2. Objetivos secundarios

- Generar una propuesta curricular que permita establecer pautas y lineamientos para el desarrollo de un curso sobre GSD con estudiantes ubicados en diferentes zonas geográficas.
- Realizar un análisis exhaustivo de las prácticas de metodologías ágiles que fomenten el desarrollo de las habilidades suaves en los estudiantes.
- Adaptar técnicas de las metodologías ágiles que fomenten el desarrollo de las habilidades suaves, enfocadas a la gestión de proyectos en entornos de GSD.
- Diseñar e implementar una plataforma tecnológica con el objetivo de evaluar, en un entorno educativo real, el marco de referencia desarrollado.
- Desarrollar un estudio cualitativo/cuantitativo que obtenga las opiniones de los estudiantes y profesores, así como para medir el rendimiento académico y las habilidades de comunicación y trabajo en equipo de los estudiantes con la intención de comprobar que la solución propuesta en la tesis contribuye al desarrollo de las habilidades suaves, necesarias en entornos de GSD.

1.7. Aproximación metodológica para la solución

Tomando en cuenta lo anteriormente expuesto, esta tesis es desarrollada considerando dos principales aportaciones: el marco de referencia en sí, y la plataforma tecnológica que permita su evaluación en un entorno real. Por consiguiente, el enfoque metodológico a seguir contempla las técnicas, procedimientos, y métodos mencionados en las siguientes secciones.

1.7.1. Diseño del marco de referencia

El marco de referencia está compuesto por bases teóricas y pedagógicas que permiten establecer las pautas y lineamientos para la realización de un curso sobre el GSD a nivel licenciatura. Dicho marco es diseñado considerando las recomendaciones establecidas por la teoría constructivista, que define al aprendizaje como un proceso activo, en el cual los estudiantes son constructores de su propio conocimiento y los profesores son la guía que los ayuda durante el proceso de aprendizaje. Derivado de esta teoría surge el enfoque de Aprendizaje Basado en Proyectos (ABP) que, en términos generales, permite a los estudiantes aprender a través de sus acciones y de la aplicación de su conocimiento teórico. Con este método de enseñanza se pretende diseñar un marco de referencia que permita al estudiante realizar actividades realistas, similares a las que los profesionistas cumplen en entornos reales de trabajo. Es por esto que esta tesis es diseñada haciendo uso del ABP para delinear las actividades académicas que son propuestas en el marco de referencia. Siguiendo este esquema, este marco permite diseñar y organizar un curso entre dos universidades teniendo como elemento central el desarrollo de un proyecto de software (real o ficticio) siguiendo el enfoque de GSD (véase Figura 1.1). Esto facilitará que los estudiantes

adquieran experiencia y desarrollen las habilidades necesarias para desarrollar correctamente el proyecto.

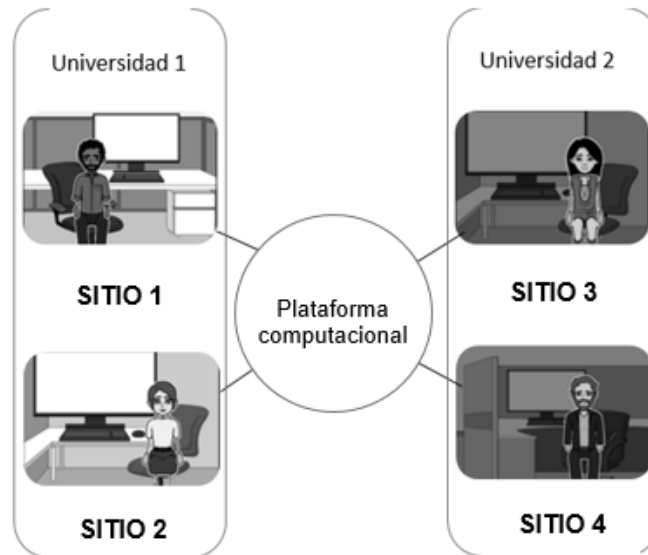


Figura 1.1. Esquema del paradigma de GSD desde la perspectiva del marco de referencia propuesto

Se requiere también de la revisión de metodologías enfocadas al diseño curricular con el objetivo de formular una propuesta adecuada a la problemática identificada. Por ejemplo, los lineamientos establecidos en la metodología propuesta por O'Neill (2015) para el diseño curricular en la educación superior definen un proceso circular con componentes que no son mutuamente excluyentes y que se relacionan directamente entre sí. Por lo tanto, se crea un marco de referencia que proporcione un equilibrio entre teoría y práctica para el diseño de los cursos, y que considere al entorno (e.g., social, físico, económico, cultural) en el cual se desarrollará el curso. De modo que se considera seguir las siguientes actividades:

- **Análisis de necesidades y filosofía educativa:** El diseño curricular establece, por un lado, que se deben tomar en cuenta las necesidades que originan el desarrollo del curso y, por otro, un conjunto de valores y creencias a las que aspiran los estudiantes de licenciatura. En otras palabras, el marco de referencia debe desarrollar una filosofía educativa que considere las necesidades educativas del programa (e.g., Ingeniería en Computación, Licenciatura en Informática) y los intereses de los estudiantes.
- **Diseño del modelo curricular:** Se requerirá la exploración y análisis de planes de estudio, ya definidos, que pudieran ser tomados en cuenta para su integración con el marco de referencia. Esta exploración permitirá definir un modelo de referencia que se adapte a las necesidades y filosofía educativa establecidas en el punto anterior.
- **Establecimiento de objetivos y resultados:** Durante esta actividad se definirán las habilidades y competencias que se esperan inculcar en los estudiantes, así como los objetivos educativos y resultados esperados de la implementación del marco de referencia.
- **Organización y estructura del programa:** Se deberá establecer una relación entre los componentes definidos en el marco de referencia considerando su alcance, secuencia, continuidad, integración, articulación y balance en relación con el programa al que será integrado.
- **Diseño de estrategias de enseñanza, aprendizaje, y evaluación:** En esta actividad se deberá diseñar un plan coherente que integre los enfoques de enseñanza, aprendizaje, evaluación, y

realimentación que estén relacionados con la filosofía educativa diseñada en la primera actividad.

- Diseño del módulo: En este punto se integrarán finalmente todos los componentes obtenidos de las fases anteriores, para ello se debe de tomar en cuenta la implementación del enfoque ABP en una plataforma tecnológica que sirva de soporte a las clases tradicionales.

Al mismo tiempo se pretende que los estudiantes adquieran la habilidad que les permita identificar el momento oportuno para realizar una interacción con sus compañeros de equipo que, en términos generales, se denomina “comunicación informal”. Este tipo de comunicación no está planificada y puede darse en cualquier momento durante el desarrollo del proyecto. Para esto se implementa el concepto de Esferas de Trabajo Colaborativo (CWS, por sus siglas en inglés) propuesto por Palacio, Vizcaíno, Moran, y Gonzáles (2011), que se basa en el concepto de Esferas de Trabajo (WS, por sus siglas en inglés). Las WS describen las unidades de trabajo que utilizan las personas para organizar y definir sus actividades y responsabilidades haciendo referencia a tareas cortas, como arreglar componentes de software, dar mantenimiento a equipo, etc. Sin embargo, este enfoque está limitado al trabajo individual. En este contexto, CWS combina las WS y los Espacios de Colaboración Potenciales (PCA, por sus siglas en inglés) que sugieren monitorear el contexto para determinar un momento de colaboración entre el equipo. Esta combinación permite a los trabajadores detectar, identificar, o crear oportunidades para la interacción, comunicación, y colaboración con sus compañeros, basándose en la información de sus WS. No obstante, para el desarrollo de esta tesis se plantea añadir una capa más al modelo conceptual de CWS con el objetivo de que un estudiante, en el rol de líder de proyecto, analice las interacciones realizadas entre los miembros del equipo de trabajo, establezca y dirija reuniones informales, y mantenga al grupo bajo una comunicación constante (véase Figura 1.2). Cabe señalar que este líder de proyecto puede monitorear el rendimiento de su equipo con el objetivo de gestionar eficientemente el desarrollo del proyecto y estar mejor comunicado con el mismo. La integración de este modelo se lleva a cabo como una actividad de enseñanza que será especificada en la actividad de “Diseño de estrategias de enseñanza, aprendizaje, y evaluación”, definida anteriormente.

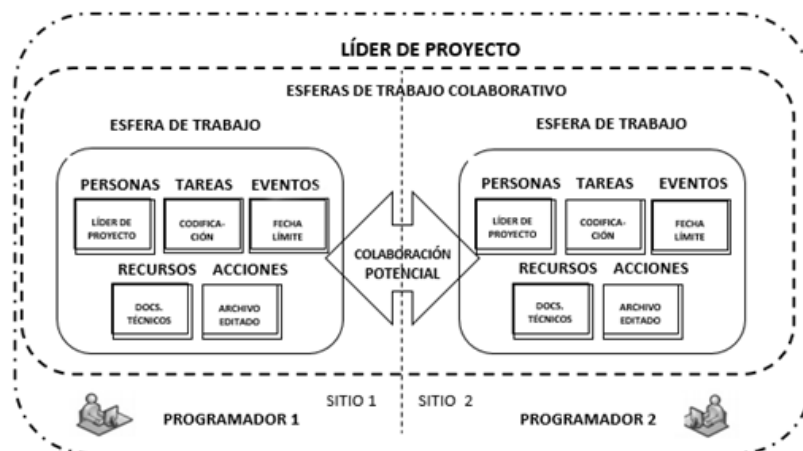


Figura 1.2. CWS enfocado a líder de proyecto (modificado de Palacio et al., 2011)

1.7.2. Diseño y construcción de la plataforma tecnológica

Como se ha mencionado a lo largo de este documento, esta tesis también considera el diseño y la construcción de una plataforma computacional que permita la evaluación, en un entorno

académico real, del marco de referencia creado mediante la organización, implementación, gestión, y evaluación de un curso sobre el GSD. Esta plataforma permite conducir un curso a través de cuatro fases principales:

- Fase 1 – Configuración del curso: Durante esta fase se establecerán los lineamientos para organizar a los equipos de trabajo, establecer los horarios de trabajo, los medios de comunicación, los roles de los integrantes del equipo, las horas dedicadas al proyecto, y demás cuestiones relacionadas con el inicio del curso.
- Fase 2 – Configuración del entorno: En esta fase se determinarán y configurarán los lugares de trabajo, los equipos, el software y hardware de comunicación, etc.
- Fase 3 – Desarrollo de la solución: Durante esta fase los equipos de estudiantes desarrollarán el proyecto siguiendo las técnicas recomendadas por los métodos ágiles que serán implementados (e.g., *Scrum*, *eXtreme Programming*, etc.).
- Fase 4 – Evaluación: Esta fase evaluará los lineamientos establecidos para la valoración del curso (e.g., desempeño de los estudiantes, penalizaciones, métodos y criterios de evaluación, entre otros) de forma paralela con la Fase 3, efectuando evaluaciones constantes y generando realimentación para los estudiantes.

El diseño y construcción de la plataforma seguirán los lineamientos del modelo incremental, que combina elementos del modelo lineal secuencial (aplicado repetidamente) con la filosofía interactiva de construcción de prototipos, que a partir de un conjunto de requisitos funcionales logra la implementación a través de algún lenguaje de programación. Es así como se utiliza el modelo incremental para el desarrollo de software propuesto por Mills, O'Neill, Linger, Dyer, y Quinnan (1980), como una forma de reducir la repetición del trabajo en el proceso de desarrollo de la plataforma y proporcionar oportunidades para concretar las decisiones sobre los requisitos detallados, hasta que se adquiera cierta experiencia con su uso. Finalmente, la tesis proporcionará la información general sobre el diseño del soporte computacional de manera que ésta pueda servir como referencia a otros proyectos similares.

Por tanto, la herramienta computacional permite facilitar la organización y control de los equipos, gestionar los avances del proyecto, gestionar las reuniones, realizar las evaluaciones, y demás.

1.7.3. Experimentación

Una vez que la plataforma tecnológica haya sido terminada, es necesario diseñar un caso de estudio que contemple la participación de cinco universidades en la realización de un curso de GSD que siga los lineamientos del marco de referencia y que haga uso de la plataforma desarrollada. Para esto, se imparte primero un curso teórico acerca del GSD, y posteriormente se desarrollan proyectos de software siguiendo este enfoque y aplicando los conceptos teóricos impartidos durante el curso.

De manera general se considera que una vez que se termine el curso teórico los participantes en el estudio formen equipos de trabajo que son integrados por un líder de proyecto y tres estudiantes más con el rol de programadores. De acuerdo con la teoría establecida por Göll y Nafalski (2007) los grupos colaborativos deben ser pequeños, normalmente no superan los cinco miembros, y se pretende que sean propicios para lograr un aprendizaje exitoso que pueda demostrarse mediante la adquisición de conocimientos y habilidades o por la realización de un conjunto de tareas. Cada equipo tiene como objetivo desarrollar un proyecto de software siguiendo

el paradigma de GSD. Para ello, los profesores desarrollan el rol de propietario del producto para el cual se debe desarrollar el proyecto siguiendo un enfoque de desarrollo ágil, utilizando prácticas de la metodología *Scrum*. La elección del enfoque ágil se fundamenta principalmente en los argumentos de Holmström, Fitzgerald, Ågerfalk, y Conchúir (2006) que afirman que un enfoque ágil ayuda a reducir los problemas relacionados con las distancias temporal, geográfica, y socio cultural, presentes en los entornos de GSD y, como consecuencia, mejoran la comunicación y el trabajo en equipo. Una vez que se haya identificado el proyecto a desarrollar, éste es asignado al equipo a través de la plataforma tecnológica para que posteriormente el equipo realice la planificación y el desarrollo de éste. La plataforma permite a los profesores monitorear el trabajo de los equipos y a los estudiantes les sirve de apoyo en la comunicación y colaboración, de esta manera, la planificación del proyecto y de los *sprints* es realizada a través de reuniones en la plataforma. Cabe mencionar que es tarea del profesor estar presente en cada reunión del equipo para observar el trabajo de los estudiantes. En este sentido, si algún integrante del equipo GSD tuviera alguna complicación en relación con su participación en el proyecto, ésta debe ser resuelta durante la reunión con la participación de todos los integrantes, fomentando así el trabajo colaborativo.

Por otro lado, con el objetivo de evaluar la efectividad del modelo de referencia se medirán cinco dimensiones a lo largo del caso de estudio: motivación, satisfacción, rendimiento académico de los estudiantes, comunicación, y trabajo en equipo. Teniendo este objetivo en mente, es necesario diseñar una encuesta para evaluar la efectividad del modelo de referencia y el enfoque de ABP. Dicha encuesta será diseñada siguiendo una escala tipo Likert de 4 puntos: “Totalmente en desacuerdo” (TD) = 1, “En desacuerdo” (D) = 2, “De acuerdo” (A) = 3, y “Totalmente de acuerdo” (TA) = 4. Aunado a lo anterior, es necesario diseñar y aplicar un examen pre y post al uso de la plataforma tecnológica con el objetivo de evaluar la mejora en el rendimiento académico de los estudiantes. De manera adicional, se evaluará la percepción de los profesores que participen en el uso de la plataforma tecnológica.

1.8. Estructura de la tesis

La estructura del documento de tesis se detalla a continuación.

El Capítulo 2 presenta el marco conceptual sobre el enfoque GSD y expone un análisis sobre propuestas similares a la presentada en esta tesis, con el objetivo de realizar un *benchmarking* que considere las mejores características exploradas por investigaciones anteriores.

En el Capítulo 3 se describen los elementos de la plataforma computacional y de la propuesta curricular que da soporte a la tecnología propuesta.

El Capítulo 4 presenta la validación de la solución propuesta mediante el diseño de un caso de estudio entre, al menos, dos universidades (e.g., Universidad Tecnológica de la Mixteca y NovaUniversitas).

El Capítulo 5 presenta las conclusiones y líneas futuras sobre este trabajo.

El Anexo A presenta una lista de acrónimos que se mencionan a lo largo de la tesis.

Por último, se presentan las referencias bibliográficas utilizadas en el desarrollo de esta tesis.

2. Marco teórico

El concepto de GSD tiene su origen en la década de los 90's cuando la industria de software comenzó a practicar la subcontratación de los sistemas de información, no obstante, el fenómeno de la subcontratación no era algo nuevo. En la década de los 60's, cuando las computadoras eran extremadamente caras y físicamente enormes, éstas requerían de grandes espacios para su instalación y de una inversión económica importante para poder adquirirlas. Además, en aquel entonces las computadoras necesitaban ser ubicadas en un entorno controlado para poder funcionar, lo cual aumentaba la inversión de las empresas. De modo que para evitar invertir en hardware muchas empresas contrataban el servicio de procesamiento de datos (Lee, Huynh, Chi-wai, y Pi, 2000).

Mientras que en la década de los 60's el mayor problema de los Sistemas de Información (SI) fue la adquisición del hardware, en los años 70's fueron los gastos del desarrollo del software lo que preocupaba más a las empresas. Ya en esta década se comenzaron a desarrollar sistemas cada vez más complejos y extensos, como sistemas de gestión de bases de datos y monitores de comunicación. Debido a lo anterior, la demanda de software se incrementó al mismo tiempo que era necesario más personal que se dedicara a desarrollar estos sistemas, por tal motivo comenzaron a surgir los contratos de programación como una forma de subcontratación que ayudaba a las empresas a desarrollar sus sistemas.

Con el surgimiento de las minicomputadoras y computadoras personales de bajo costo en los años 80's, la subcontratación de los servicios de procesamiento de datos se vino abajo y cada empresa comenzó a comprar equipo de cómputo, software de sistemas, software de aplicación, etc., organizando así su propia infraestructura de hardware y software y dejando de contratar servicios externos.

A principios de los años 90's surge nuevamente el interés por la subcontratación de servicios, pero a diferencia de la década de los 60's, en la cual las empresas subcontrataban el servicio de procesamiento de datos que se brindaba desde un lugar externo, en los años 90's los servicios de subcontratación se enfocaban en brindar un servicio en la misma ubicación del cliente. En este tipo de subcontratación, denominada completa, el personal que brindaba el servicio era parte del proveedor y el cliente pagaba por dicho servicio. Por otro lado, la subcontratación parcial se caracterizó por transferir al menos un segmento de la responsabilidad del SI de una organización a un proveedor de servicios externo, de esta manera se subcontrataban servicios de gestión de telecomunicaciones, integración de sistemas, desarrollo de aplicaciones, entre otros (Lee et al., 2000).

Los años 90's significaron un cambio en la manera en que las empresas diseñaban y realizaban sus estrategias de negocio. En las décadas de los 70's y 80's se invertían grandes cantidades de dinero en la adquisición de hardware y software de aplicación, sin planes bien organizados y

considerando que las Tecnologías de la Información (TI) por sí mismas representaban una ventaja competitiva frente a otras empresas. Mientras que en la década de los 90's las empresas se dieron cuenta de que las TI por sí solas no les proporcionaban ninguna ventaja respecto a sus competidores, sino que la ventaja era consecuencia de la manera en que hacían uso de éstas. Debido a esto, tanto las empresas pequeñas que no contaban con un departamento dedicado al desarrollo de SI, como las grandes empresas que sí contaban con éste, comenzaron a externalizar sus funciones de SI por medio de la subcontratación, generando así una relación de socios entre el proveedor y la empresa más que una relación de cliente-proveedor.

En este contexto, el desarrollo de software se hizo global en la década de los 90's como consecuencia de la revolución de las computadoras y los problemas que se experimentaban en el desarrollo de software, tales como: presupuestos ajustados, escasez de recursos, y escasez de tiempo. Estos factores motivaron a las empresas a buscar socios o establecer sitios de desarrollo en diferentes países. Como resultado de lo anterior, muchas empresas se unieron con otras para formar una empresa conjunta y trasladaron sus centros de desarrollo a países de bajo costo operativo (Šmite, Wohlin, Gorschek, y Feldt, 2010).

Una de las características más significativas de las TI a principios del siglo XXI es la globalización⁴, la cual ha permitido la conectividad e integración en áreas como la política, cultural, social, económica y de sistemas tecnológicos entre naciones, empresas, individuos, etc., alrededor del mundo. Es así como la globalización afecta al mundo en diferentes formas, incluyendo el surgimiento de nuevos mercados de producción y consumo, nuevos mercados financieros, nuevas formas de desarrollar SI, entre otros. Particularmente, ésta ha impactado directamente en la forma en cómo se desarrollan los SI puesto que ha permitido que muchos sistemas, que tradicionalmente se desarrollaban por un equipo de desarrolladores en un mismo sitio, sean ahora desarrollados por otros desarrolladores ubicados en empresas y países diferentes, fomentando aún más el desarrollo global (Cho, 2007).

Durante la primera Conferencia Internacional sobre Desarrollo Global, organizada en el 2006 por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés), los investigadores y profesionales reconocían que desarrollar software más allá de las fronteras de donde se ubicaba una empresa se estaba convirtiendo en una ventaja competitiva dentro de la industria del software, pero que también la globalización del desarrollo de software creaba desafíos para la IS debido al impacto de las diferentes zonas horarias, la diversidad cultural, y la comunicación o la distancia. Por consiguiente, el objetivo de la conferencia era reunir a profesionales e investigadores para llevar a cabo una exploración del estado del arte y práctica del desarrollo global. A partir de entonces el interés por GSD ha ido en aumento.

En este sentido, el crecimiento de GSD ha ido de la mano con los avances tecnológicos en la comunicación, tales como el uso del correo electrónico, los mensajes instantáneos, y los dispositivos de comunicación. Aunado a lo anterior, la disponibilidad de ingenieros de software a bajo costo en lugares como América Latina, el lejano oriente y Europa del este, y el deseo de las empresas de reducir sus costos de operación y obtener ventajas al establecer operaciones cerca de los mercados emergentes han hecho que cada vez más organizaciones decidan trabajar siguiendo el enfoque del GSD (Noll, Beecham, y Richardson, 2010). De manera similar, Cho (2007) identificó cinco factores que motivan a las empresas a trabajar bajo el enfoque GSD:

⁴ De acuerdo con la Real Academia de la Lengua Española, la *globalización* es el proceso por el que las economías y mercados, con el desarrollo de las tecnologías de la comunicación, adquieren una dimensión mundial, de modo que dependen cada vez más de los mercados externos y menos de la acción reguladora de los gobiernos.

- Menores costos de desarrollo: Dado que al contratar programadores en países donde los salarios son más reducidos las empresas pueden abaratar los costos de desarrollo.
- Posibilidad de contratar programadores hábiles y capacitados: Dado que, mediante el enfoque global, la contratación de personal en diversos países, como la India y China, ha generado especial atención en la formación de ingenieros de software cada vez mejor preparados para trabajar bajo el enfoque del GSD, de tal forma que las empresas contratantes aprovechen sus capacidades al máximo.
- Capacidad para crear empresas y equipos virtuales: Dado que las empresas tienen la posibilidad de crear, en un corto periodo de tiempo, equipos y empresas virtuales alrededor del mundo para aprovechar las oportunidades del mercado, formando equipos virtuales de desarrolladores bien calificados.
- Jornadas laborales extensas: Dado que el GSD permite usar la diferencia horaria para lograr jornadas amplias de trabajo siguiendo el enfoque denominado “seguir el sol”, el cual permite generar jornadas de trabajo de hasta 24 horas, logrando un mejor rendimiento del personal en el desarrollo.
- Ventajas de negocio: Dado que las empresas pueden obtener una ventaja respecto a sus competidores al extender sus fronteras y acercarse a nuevos mercados.

Ahora bien, para que las empresas trabajen bajo el enfoque de GSD deben crear equipos virtuales de desarrolladores que estén ubicados en diferentes zonas geográficas, a diferencia del enfoque de desarrollo tradicional en donde los equipos están ubicados en un mismo lugar. Los equipos virtuales al igual que los equipos tradicionales se conforman de un grupo de desarrolladores que trabajan en conjunto para lograr las mismas metas y objetivos e interactúan a través de tareas interdependientes, pero están ubicados en diferentes zonas geográficas y diferente zona horaria, además de que posiblemente tengan diferentes culturas. Estos equipos trabajan en entornos multiculturales, en los cuales existe la posibilidad de que cada integrante del equipo tenga una cultura y una lengua distinta al resto de sus compañeros. Igualmente, la comunicación entre el equipo se da normalmente a través de medios electrónicos, lo cual dificulta la comunicación entre el equipo (Noll et al., 2010).

Así mismo, Ågerfalk, Fitzgerald, Holmström, y Conchúir (2008) mencionan algunos beneficios que pueden obtenerse al utilizar un enfoque de GSD:

- Las empresas de desarrollo de software contratan a programadores en lugares donde los salarios son más reducidos y por consiguiente pueden abaratar los costos de desarrollo.
- Las empresas tienen la posibilidad de contratar ingenieros de software con diversas habilidades y hacerlos trabajar en equipo para obtener mejores beneficios en el desarrollo de software.
- Las empresas pueden realizar jornadas de trabajo más extensas debido a la diferencia de horarios entre los lugares de desarrollo, pudiendo lograr hasta jornadas de 24 horas.
- Las empresas pueden tener equipos de desarrollo en la ubicación de sus clientes, de esta manera, pueden conocer y explorar otros mercados.
- Las empresas pueden aprovechar la innovación que surge de unir a personas de diferentes sitios, con habilidades particulares y con una forma de trabajo diferente, así mismo pueden

compartir las mejores prácticas que surgen de la colaboración entre los integrantes de los equipos virtuales.

- La distribución geográficamente organizada de las unidades de desarrollo implica un cierto grado de autonomía para cada equipo que trabaja en dicha unidad.
- Como se mencionó anteriormente, el GSD utiliza tecnologías como el correo electrónico, mensajes instantáneos, entre otros, que normalmente dejan un historial escrito de la comunicación, esto proporciona una mayor trazabilidad y rendición de cuentas independientemente de la ubicación geográfica de los desarrolladores.
- Los equipos de GSD tienen un mayor enfoque en la documentación para facilitar su comunicación. La información se documenta y se distribuye electrónicamente a través de los diferentes sitios.

No obstante, existen diversas barreras que una empresa debe superar para alcanzar los beneficios mencionados. De acuerdo con Cho (2007), la comunicación, coordinación, y control son los mayores retos a los que se enfrentan los equipos de GSD. Por otra parte, es bien sabido que establecer normas bien definidas de comunicación en los equipos de desarrollo es difícil y lograr una buena comunicación en equipos de GSD es más complicado aún, debido principalmente a las distancias geográficas. Por tal motivo, los integrantes de los equipos no son capaces de establecer una comunicación informal en donde puedan tener charlas frente a frente. Debido a lo anterior, diversos investigadores y profesionales de la industria han sugerido el uso de diversas tecnologías como el correo electrónico, escritorios remotos, mensajes instantáneos, protocolos de transferencia de archivos (FTP, por sus siglas en inglés), redes virtuales privadas (VPN, por sus siglas en inglés), videoconferencias, entre otros, con el objetivo de mejorar la comunicación entre los equipos de GSD. Sin embargo, para el correcto funcionamiento de dichas herramientas es necesario que los sitios donde se encuentren los equipos cuenten con una conexión a internet. Dicha conexión debe ser lo suficientemente estable para que la comunicación entre los sitios no falle.

Por otro lado, los problemas en la coordinación y control en GSD surgen cuando se divide y organiza el trabajo de los equipos. En este sentido, la gestión del proyecto en los entornos de GSD se vuelve más compleja que en los entornos tradicionales. De acuerdo con Noll et al. (2010), la esencia de una gestión eficaz es la coordinación y el control. Sin embargo, la distancia que se genera en los entornos de GSD afecta directamente en la comunicación y cooperación entre los equipos involucrados, de igual manera, la visibilidad y el conocimiento del proyecto se ven afectados por la distancia. Así pues, la coordinación y control de los proyectos de GSD se ven afectados directamente por los puntos mencionados anteriormente, provocando así la falta de cooperación entre los integrantes de los equipos. De manera similar, gestionar las habilidades de los integrantes del equipo en entornos de GSD resulta más complicado debido a las diferencias de lenguaje y cultura originadas de la unión de ingenieros de software de diferentes países, además la transferencia del conocimiento del proyecto entre los integrantes del equipo suele ser escasa principalmente porque éstos no se sienten parte de un equipo al estar geográficamente distanciados.

Con respecto a lo anterior, Noll et al. (2010) clasificaron en ocho categorías las barreras que las empresas de desarrollo de software deben superar para adoptar el enfoque de GSD:

- La *distancia geográfica* introduce diversas barreras con relación a la colaboración entre los integrantes del equipo, la principal es la falta de encuentros informales que provee no solo la

oportunidad de intercambiar conocimiento del proyecto, sino también permite generar relaciones interpersonales.

- La *distancia temporal*, derivada de las diferencias geográficas, ocasiona que haya pocas horas en las que los equipos puedan participar en reuniones asíncronas. Así mismo, provoca retardos en la transferencia del conocimiento.
- La *cultura e idioma*, por otro lado, dificultan no solo la comunicación, sino también la elección de los medios de comunicación. En este contexto, el inglés se toma como lengua común en la IS, por lo tanto, aquellas personas que no dominan el inglés prefieren tener comunicación por medio de textos y correo electrónico en lugar de llamadas o videoconferencias. Aunado a lo anterior, la cultura influye en la interpretación de la comunicación principalmente por las diferentes interpretaciones que puede tener cierta acción en diferentes culturas.
- El *temor y la confianza* se ven afectados debido a la falta de reuniones cara a cara que provoca que los integrantes del equipo no generen relaciones interpersonales que mejoren la confianza entre compañeros. Por otro lado, cada uno de los integrantes del equipo se siente inseguro acerca de la estabilidad de su trabajo, este fenómeno se presenta con frecuencia en el personal que trabaja en la empresa y que cree que al implementar el GSD será despedido para contratar a otros desarrolladores de otros países.
- Los *problemas derivados de la estructura organizativa* se presentan dado que el GSD puede dividir los grupos en uno con el poder de tomar decisiones y ejecutar el proyecto y otro con el conocimiento de las necesidades de los *stakeholders*.
- Los *problemas del proceso* se originan debido a la falta de comunicación que provoca que un mismo proceso pueda ser llevado a cabo de diferente manera en cada uno de los sitios de desarrollo.
- Los *problemas derivados de la infraestructura* se deben principalmente a que el GSD utiliza herramientas como las videoconferencias para realizar la comunicación entre los equipos; por lo tanto, cualquier falla en el hardware o software de comunicación provoca que las reuniones no se lleven a cabo.
- Finalmente, los *problemas derivados de la arquitectura del proyecto* se originan debido a que las arquitecturas que requieren múltiples sitios para implementar un cambio aumentan el tiempo requerido para completar dicho cambio. Así mismo, una arquitectura inestable que cambia mientras el trabajo de desarrollo está en curso puede causar confusión entre los equipos en cuanto a cuáles son sus responsabilidades.

En consecuencia, el GSD presenta aún una serie de problemas y retos a resolver que dificultan su implementación, y obtener los beneficios que este enfoque brinda continúa siendo un reto en la actualidad. No obstante, el GSD es un paradigma dinámico que se encuentra en constante evolución ya que, de acuerdo con Vizcaíno et. al. (2015), el problema de la distancia geográfica en entornos globales se ha reducido considerablemente gracias al uso de las tecnologías adecuadas y de la mejora del proceso de software en entornos deslocalizados. De igual manera, las diferencias lingüísticas y culturales, además de la distancia geográfica, ya no son consideradas factores críticos en la implementación del GSD, en cambio, la motivación personal, las habilidades del personal, y disponer de funciones y responsabilidades bien definidas son considerados actualmente factores

críticos en esta área. Con base en lo anterior, estos mismos autores afirman que el GSD ha comenzado a alcanzar un cierto grado de madurez que, al mismo tiempo, también introduce nuevos desafíos en diferentes líneas de interés como:

- **Procesos para desarrollo y gestión:** Los procesos que tradicionalmente se aplican en entornos tradicionales de desarrollo no pueden ser correctamente aplicados en entornos de GSD debido a las particularidades de este enfoque, por tal motivo, la tendencia en esta línea de investigación es la aplicación de métodos ágiles de desarrollo, como *Scrum*, adaptados a entornos globales.
- **Gestión de proyectos en entornos de GSD:** La gestión de proyectos es una línea de interés que cobra relevancia en estos entornos de desarrollo. En este contexto, la mayor parte de la investigación existente se ha enfocado en una correcta estimación de esfuerzo y costos del proyecto, considerando estrategias para realizar de manera correcta la asignación de tareas en las distintas ubicaciones de desarrollo.
- **Equipos de trabajo:** Los equipos de trabajo deben mejorar los aspectos de comunicación, coordinación y control, para ello es importante que los integrantes de los equipos tengan una formación correcta que les permita mejorar sus capacidades en el ámbito del GSD.

2.1. Aspectos relevantes sobre la enseñanza de GSD en entornos universitarios

Como se ha mencionado anteriormente, uno de los principales actores en el éxito de los proyectos de GSD es el personal que trabaja bajo este enfoque. Por lo tanto, proporcionar una formación adecuada a los futuros ingenieros de software dentro de las aulas universitarias es una de las principales estrategias que permitirían disminuir los problemas y retos del GSD. En este sentido, Matthes et al. (2011), argumentaron que los ingenieros de software se enfrentarían a nuevos retos relacionados con el GSD en su vida laboral, por lo cual era importante mejorar la enseñanza de este enfoque en las universidades. Sin embargo, la manera de enseñar la IS sigue sin adaptarse correctamente a estos nuevos desafíos. Similarmente, Richardson, Moore, Paulish, Casey, y Zage (2007) concluyeron que era importante que los estudiantes de IS se familiarizaran con los enfoques del GSD desde su formación universitaria. Pero también destacaron que era importante que los profesores que enseñan la IS implementaran enfoques pedagógicos innovadores que proporcionen a los estudiantes una mejor educación, de tal forma que éstos puedan desenvolverse como los ingenieros “globales” de software que la industria requeriría en el futuro.

Con respecto a lo anterior, diversos líderes de proyectos de GSD afirman que los ingenieros de software recién egresados de las universidades no cuentan con las habilidades y el conocimiento necesarios para afrontar los retos del GSD, puesto que argumentan que su experiencia se limita a pequeños proyectos, principalmente porque la educación en IS no aborda generalmente los temas relacionados con el GSD (Monasor, Vizcaíno, Piattini, y Caballero, 2010). Así pues, en entornos de GSD las reuniones informales del equipo de trabajo no son muy comunes, por lo tanto, la comunicación y coordinación se convierten en tareas difíciles de realizar. Además, el uso incorrecto de las herramientas de comunicación puede generar efectos negativos en el desempeño de los equipos virtuales, tales como retardo en los tiempos de respuesta y falta de confianza entre compañeros. Por tal motivo, es importante que los estudiantes adquieran los conocimientos y las habilidades necesarias, relacionadas con la comunicación, que les permitan utilizar el lenguaje y las herramientas de manera eficiente. También es importante que los estudiantes comprendan el impacto de la diversidad cultural en el desarrollo de los proyectos globales y que el conocimiento sobre ésta puede proporcionar grandes beneficios al equipo (Richardson et al., 2007).

Por ende, es importante que los estudiantes adquieran las habilidades necesarias para desempeñarse como profesionales capaces de afrontar los nuevos desafíos presentes en el GSD. No obstante, los temas relacionados a este enfoque de desarrollo rara vez son contemplados en los planes de estudio actuales, relacionados con la enseñanza de la IS, de las universidades. Así que preparar a los estudiantes en este campo no es fácil, ya que se requiere de nuevos contenidos teóricos y herramientas de soporte, y la coordinación efectiva entre profesores, estudiantes, y universidades. Incluso es necesario proveer a los estudiantes de experiencias reales que les permitan desarrollar habilidades duras y suaves. Como consecuencia de lo anterior, diversos investigadores han tratado de abordar el problema de la educación del GSD y han propuesto cursos académicos, entornos de aprendizaje, y herramientas computacionales de soporte. En este contexto, la revisión sistemática de literatura de Monasor et. al. (2010), sobre las tendencias educativas enfocadas a la enseñanza del GSD, clasificó a los principales trabajos realizados en esta área en: casos de estudio, experimentos, simulaciones, encuestas, y modelos no experimentales que realizan propuestas teóricas (véase Figura 2.1). Considerando lo anterior, se puede observar que la mayoría de los trabajos realizados en el área son casos de estudio con un 68% del total de propuestas encontradas por los autores. Así mismo, es importante destacar que el 76% de los trabajos fueron realizados en entornos universitarios ubicados en diferentes lugares, en los cuales la colaboración representó el elemento crucial para realizar algún proyecto de GSD. En algunos de estos proyectos se presentó la colaboración de algunas empresas con el objetivo de que los estudiantes pudieran obtener conocimiento práctico sin salir de las universidades.

De manera análoga, Clear et al. (2015) clasificaron los trabajos y propuestas en el área de la enseñanza de GSD en: investigaciones empíricas (estudios que aplicaron métodos cuantitativos y cualitativos), informes de experiencias (con un enfoque limitado en la investigación), y propuestas teóricas (marcos de trabajo y perspectivas filosóficas acerca de la enseñanza de GSD). Los hallazgos de los autores sugieren que los principales trabajos con relación a las investigaciones empíricas e informes de experiencias comprenden: casos de estudio descriptivos y exploratorios, experimentos controlados, encuestas de estudiantes y profesores, cuestionarios y entrevistas, análisis de contenidos, análisis estadísticos, entre otros. En este sentido, la Tabla 5 muestra la clasificación realizada por los autores con base en la revisión sistemática de literatura realizada.

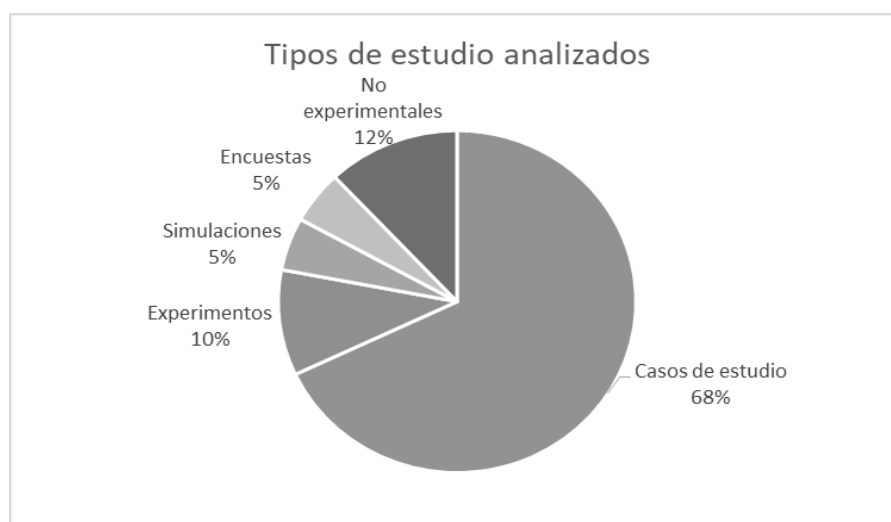


Figura 2.1. Estudios analizados por la investigación de educación de GSD (Monasor et al., 2010)

Tabla 5. Clasificación de trabajos analizados en la revisión sistemática de literatura (Clear et al., 2015).

Clasificación	# de propuestas	%
Investigación empírica	43	52.5
Informes de experiencias	34	41.6
Propuestas teóricas	5	6
Total	82	100

2.1.1. Los retos y desafíos en la educación de GSD

Con relación a los retos y desafíos en la enseñanza del GSD, el estudio de Clear et al. (2015) identificó siete retos principales a los que se enfrenta la enseñanza de éste en entornos universitarios:

- Distancia global.
- Equipos de trabajo.
- Planes de estudio / Pedagogía.
- *Stakeholders* / Roles.
- Infraestructura.
- Problemas relacionados con las personas.
- Procesos de desarrollo.

En el contexto de los entornos educativos, la distancia global hace referencia a las diferentes regulaciones institucionales, tales como: problemas de sincronización de cursos (generalmente organizados con una duración de 4 a 5 meses), los diferentes esquemas y métodos de evaluación, las expectativas y metas del curso, y las diferencias en el nivel de conocimiento y habilidad de los estudiantes participantes. Así, los problemas relacionados con la distancia global pueden clasificarse en:

- Problemas relacionados con la distancia cultural: La distancia cultural es definida considerando las diferencias en los estilos de comunicación de los estudiantes, normas culturales, creencias étnicas y religiosas, y el entendimiento de la equidad de género.
- Problemas relacionados con la distancia geográfica: Estos problemas hacen referencia al impacto que la distancia geográfica tiene sobre la comunicación entre los equipos virtuales.
- Problemas relacionados con las diferencias lingüísticas: El lenguaje ha sido considerado un problema en la comunicación entre equipos de GSD, debido a que muchos estudiantes no dominan el inglés (generalmente utilizado como lengua base en la IS), provocando que éstos no se integren y/o relacionen con sus compañeros por temor de no ser comprendidos o aceptados.
- Problemas relacionados con la distancia organizacional: Organizar cursos entre dos o más instituciones representa un serio problema en la enseñanza del GSD, dadas principalmente las diferencias en los horarios, cambios continuos de las reglas, diferentes regulaciones institucionales, diferentes puntos de vista y expectativas por parte de profesores, diferencias en los planes de estudios, diferentes formas de evaluación e incompatibilidad en vacaciones y días festivos de cada sitio.

- Problemas relacionados con las habilidades de los estudiantes: El GSD permite la participación de estudiantes con diferente nivel de conocimiento. Este hecho sugiere problemas para los profesores que deberán balancear los equipos de acuerdo con el nivel de conocimiento de los estudiantes. Además, los profesores deberán gestionar adecuadamente las habilidades de los participantes para obtener mejores beneficios.
- Problemas relacionados con la distancia temporal: Al implementar un curso de GSD entre dos instituciones es muy probable que éstas estén ubicadas en diferentes zonas horarias, como consecuencia, programar las reuniones entre los equipos virtuales y coordinar las actividades de cada equipo son tareas complejas.

Por otro lado, el trabajo en equipo en entornos de GSD se ve afectado por los problemas ocasionados en las distancias tanto temporal como cultural. En el contexto de los entornos educativos, los problemas relacionados con el trabajo en equipo hacen referencia a la sinergia (cohesión, integración y cooperación entre equipos globales) y al problema de la asignación de tareas. La sinergia refleja la dificultad de los estudiantes para generar relaciones de trabajo efectivas con sus compañeros, mientras que los problemas con la asignación de tareas exponen la dificultad de los profesores y estudiantes al crear equipos de trabajo balanceados con relación a sus habilidades (Clear et al., 2015).

Con relación a los problemas en la definición de roles de los *stakeholders*, el estudio menciona que la educación de GSD comprende cuatro roles principales: profesores, estudiantes, clientes, y universidades. Cada uno de los participantes en la educación del GSD enfrenta un cierto grado de complejidad y retos al momento de realizar sus actividades. Con relación a lo anterior, los profesores tienen el reto de gestionar los equipos de estudiantes globales, así mismo, se enfrentan a la dificultad de planificar y coordinar las actividades del curso. Por otro lado, los principales problemas a los que se enfrentan los estudiantes son la gestión de sus habilidades, su preparación, y no perder la motivación y enfoque del curso. Otros de los problemas que tienen los estudiantes son: falta de atención a los detalles del proceso distribuido, incapacidad para administrar los plazos y las actividades, falta de interés por participar en las reuniones de equipos, y falta de conocimiento sobre las herramientas de comunicación, entre otros. Por otra parte, los clientes en proyectos académicos de GSD son por lo regular un agente externo al entorno académico o bien un equipo de estudiantes que asuma este rol. En ambos casos, los autores del estudio afirman que el rol de cliente agrega complejidad para los estudiantes, debido a que éstos muchas veces asumen un comportamiento arrogante y los estudiantes son incapaces de gestionar este tipo de clientes. Más aún, cuando el cliente es externo, éste suele no integrarse de manera constante al desarrollo del proyecto y en muchas ocasiones no responde a las preguntas de los estudiantes o bien tarda mucho tiempo en responder. Finalmente, las universidades deben realizar funciones administrativas y técnicas para poder establecer cursos de GSD, lo anterior agrega complejidad para los profesores quienes deben gestionar y planificar las actividades.

De acuerdo con Beecham, Clear, y Noll (2017) cada uno de los *stakeholders* participantes en los cursos de GSD asumen responsabilidades que pueden llevar al éxito o fracaso de éste. Los autores afirman que los profesores llevan a cabo un conjunto complejo de responsabilidades como la gestión de estudiantes, actividades, evaluaciones e incluso toman parte en la coordinación de actividades entre las universidades. Así mismo, el profesor puede tener tres niveles de participación en el curso de GSD: profesor principal (i.e., generalmente responsable de la gestión del curso en varios sitios); miembro del grupo de profesores (i.e., grupo de profesores responsables de cada grupo distribuido de estudiantes); y profesor (i.e., profesor responsable de un grupo de estudiantes).

En cuanto a los estudiantes, éstos asumen diversas responsabilidades de acuerdo con el avance del proyecto. No obstante, los clientes suelen asumir un rol pasivo en el desarrollo del curso. Por lo tanto, es importante destacar que, de acuerdo con el estudio, existe una serie de problemas y conflictos entre los roles durante los cursos de GSD y, que los estudiantes enfrentan una serie de problemas cuando se les asignan responsabilidades de gestión y liderazgo.

Como se ha mencionado anteriormente, la enseñanza del GSD involucra la participación de diferentes *stakeholders*, por lo que es importante destacar los problemas relacionados con la participación de las personas en el proceso de enseñanza de GSD:

- **Estrés:** Regularmente, los estudiantes no están acostumbrados a trabajar en proyectos relativamente grandes, por tal motivo, el involucrarse en un proyecto global implica más trabajo y esfuerzo, motivo por el cual pueden sentir estrés.
- **Motivación:** Diversos trabajos afirman que en muchas ocasiones los estudiantes no se sienten motivados, principalmente por problemas en la comunicación, diferencias en la cultura laboral, falta de compromiso de otros estudiantes o simplemente por falta de motivación desde el inicio del proyecto.
- **Confianza:** Los estudiantes que no conocen a sus compañeros o no han trabajado con otras personas suelen no tener confianza en el trabajo de sus compañeros globales.

En cuanto a los problemas de infraestructura, el estudio afirma que este tema comprende problemas relacionados con plataformas y herramientas, infraestructura de comunicación y herramientas de control de código fuente:

- **Herramientas de infraestructura:** Hace referencia a las dificultades en la instalación y dominio de tecnología de comunicación y colaboración, así mismo, a las diferencias de tecnología disponible en cada sitio.
- **Herramientas de control de versiones:** La falta del uso de herramientas de control de versiones comprende dificultades para identificar la última versión del código fuente, al mismo tiempo que hace más difícil la coordinación del proyecto.
- **Infraestructura técnica:** Hace referencia al problema de establecer un entorno tecnológico homogéneo para cada uno de los sitios, esto implica problemas desde la selección de las herramientas a utilizar como el conocimiento que cada profesor y estudiante debe poseer acerca de las herramientas.

Por otra parte, los problemas relacionados con los planes de estudio y los retos pedagógicos son:

- **Diseño del curso:** El curso de GSD debe ser diseñado de tal manera que incorpore la enseñanza de las habilidades suaves de los estudiantes, así mismo, debe tomar en cuenta para su diseño diversos factores relacionados con los estudiantes (e.g., diferente nivel de conocimiento, culturas, etc.) y factores relacionados con los profesores (e.g., realimentación, diferentes métodos de evaluación, etc.). Así mismo, el curso debe proporcionar a los estudiantes experiencias reales, herramientas y plataformas necesarias para su implementación.

- Resultados del aprendizaje: Los resultados del aprendizaje son afectados por la incapacidad de los estudiantes de realizar un plan de trabajo, por lo que tienden a enfocarse únicamente en el rol que desempeñan sin tener una visión general del proyecto.

Finalmente, los problemas relacionados a los procesos de desarrollo hacen referencia a diferentes fases del ciclo de vida del software que van desde la obtención de los requisitos hasta la integración de los componentes de software, y en cada una de ellas se encontraron problemas relacionados principalmente con la falta de conocimiento de los estudiantes y la dificultad para realizar estas tareas debido a la distancia geográfica. En este sentido, los problemas en la fase de requisitos se deben principalmente a la falta de interacción con los clientes, los problemas en el diseño del software se deben principalmente a la falta de conocimiento de los estudiantes sobre esta fase, lo cual se ve reflejado en un aumento considerable de tiempo dedicado a esta actividad, los problemas de codificación se centraron en las dificultades para comprender y modificar los sistemas. En términos generales, los problemas en el proceso de desarrollo de software son: mala calidad resultante derivada de la falta de procesos establecidos, problemas por no seguir los procesos establecidos, y cambios constantes en el proceso de desarrollo.

Por otra parte, Beecham et al. (2017) sugieren que para mejorar la enseñanza de GSD en las universidades es necesaria la participación de las empresas de desarrollo de software. Así pues, mencionan que los profesionales de la industria podrían asumir responsabilidades y roles en la enseñanza de GSD, tales como:

- Identificar proyectos GSD para los estudiantes y asumir el rol de clientes.
- Fungir como asesores de los estudiantes.
- Actuar como profesores.

Sin embargo, la participación de las empresas en los proyectos universitarios de GSD suele ser difícil de lograr debido principalmente a que las empresas temen que los estudiantes no se comprometan con el proyecto, debido a que generalmente éstos toman diferentes cursos y por tal motivo, pueden abandonar el proyecto. De manera similar, las empresas argumentan que los estudiantes generalmente no tienen las habilidades necesarias o carecen de la conciencia sobre la importancia de los problemas que enfrentan los profesionales (Beecham et al., 2017).

Con respecto a lo anterior, el estudio de Clear et al. (2015) sugiere un conjunto de recomendaciones relacionadas a la mejora de la enseñanza de GSD en las universidades. En este contexto, la Figura 2.2 resume la distribución de las recomendaciones analizadas por los autores del estudio en diferentes áreas de interés. Así mismo, la Tabla 6 muestra las principales recomendaciones realizadas en el estudio, donde la mayoría de éstas se enfocan en aspectos relacionados con la modificación de los planes de estudio y las cuestiones pedagógicas.

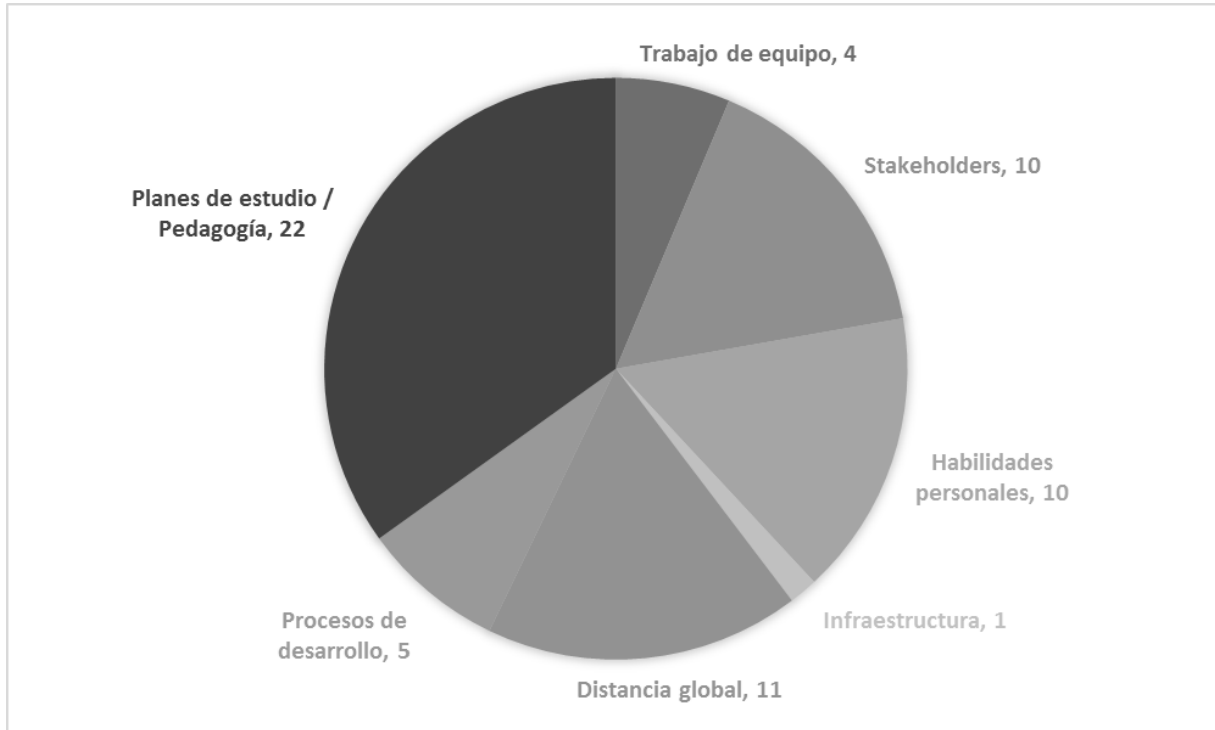


Figura 2.2. Distribución en áreas de interés de las recomendaciones del estudio de Clear et al. (2015)

Tabla 6. Principales recomendaciones acerca de la enseñanza de GSD (Clear et al., 2015)

Área de interés	Recomendaciones
Rol de los <i>stakeholders</i>	<p>Realizar concursos entre estudiantes para determinar un proyecto a realizar y así escoger al estudiante que desempeñará el rol de cliente.</p> <p>Usar simuladores en lugar de clientes reales.</p> <p>Un cliente real debe ser un representante de la empresa que pueda dedicar tiempo a los estudiantes para discutir la propuesta y el estado del proyecto.</p> <p>Asignar un profesor residente para orientar activamente al equipo local y asegurar que el proyecto cumpla con los resultados esperados por el cliente.</p>
Distancia global	<p>Armonizar los procesos entre las instituciones y establecer una línea clara de responsabilidad. Los líderes del proyecto deben aceptar, comprender, y seguir un conjunto específico de reglas y normas del curso.</p> <p>Ser flexible al aceptar diferentes reglas y hábitos.</p> <p>Los estudiantes deben ser seleccionados por el instructor en función de un perfil para garantizar un equipo equilibrado.</p>
Planes de estudio / Pedagogía	<p>Establecer el cronograma del curso incorporando plazos regulares.</p> <p>Realizar sesiones de capacitación sobre GSD pre-semestrales y generar un resumen preliminar de GSD con un enfoque en la gestión de proyectos.</p> <p>Los patrocinadores y tutores deben declarar consistentemente el objetivo principal desde el principio del curso.</p> <p>Enfocarse en el proceso antes de los temas y herramientas del proyecto.</p> <p>Los proyectos deben poderse probar con el tiempo disponible.</p> <p>Mantener el alcance del proyecto a tres meses, con resultados definidos previamente.</p>

Área de interés	Recomendaciones
	<p>Alinear el proyecto con las necesidades del patrocinador.</p> <p>Entregar un sistema de software complejo para un cliente real.</p>
Habilidades personales	<p>Hacer que los equipos locales y remotos se familiaricen entre sí lo antes posible.</p> <p>Mantener motivados a los estudiantes.</p>
Trabajo en equipo	<p>Crear una tabla de responsabilidad que incluya las tareas que los estudiantes deben completar para cumplir su rol en el curso.</p> <p>Asegurar una comunicación clara y constante para promover la visibilidad, una mejor toma de decisiones, y la comprensión de los objetivos y la sinergia del equipo.</p> <p>Comenzar la comunicación tempranamente para tener un efecto positivo en la sinergia.</p> <p>Hacer que los equipos locales y remotos se familiaricen entre sí lo antes posible.</p>
Procesos de desarrollo	<p>Administrar los conflictos de integración del software.</p> <p>Utilizar el diseño por contrato para especificar las interfaces del módulo.</p> <p>Es necesario requerir una revisión obligatoria de código antes de proceder a la implementación.</p> <p>Documentar las decisiones de diseño y la lógica para facilitar la transferencia de conocimiento a otros equipos.</p>

Aunado a lo anterior, la literatura reporta que la enseñanza del GSD se ha enfocado en preparar a los estudiantes para trabajar en proyectos desarrollados en empresas grandes, no obstante, los productos de software son producidos en su mayoría por Pequeñas y Medianas Empresas (PyMES) con menos de 250 empleados. En este sentido, las PyMES son las responsables de producir el 99.2% del software comercializado a nivel mundial y, dada la globalización actual de la industria, muchas de estas empresas trabajan con colaboradores y socios globales. En el contexto de las PyMES de desarrollo de software, los empleados generalmente asumen diferentes roles en un proyecto de software, por lo que difícilmente se enfocarán en actividades específicas. Por lo tanto, es necesario que estos empleados posean las capacidades de trabajo en equipo y liderazgo. Así mismo, los estudiantes recién egresados generalmente inician su vida laboral en este tipo de empresas. Por tal motivo, Beecham et al. (2017) sugieren que se considere preparar a los estudiantes para trabajar bajo el enfoque de GSD en el entorno de las PyMES, enfocando su preparación con relación a un conjunto de habilidades específicas relacionadas a la gestión de proyectos y liderazgo. Sin embargo, los estudiantes generalmente no cuentan con el nivel de conocimiento y habilidades necesarias para asumir responsabilidades relacionadas a la gestión de proyectos y al liderazgo de equipos de trabajo, lo cual representa un problema cuando se pretende que los estudiantes desempeñen estos roles en los proyectos de GSD, ya que, como se ha mencionado anteriormente, la gestión de proyectos y el liderazgo en entornos de GSD resultan más complejos que bajo un enfoque tradicional de desarrollo.

2.1.2. La enseñanza del GSD y la gestión de proyectos

De acuerdo con Colomo-Palacios et al. (2014), en los entornos de trabajo de GSD, el rol de los jefes de proyectos es crucial para alcanzar el éxito del proyecto, debido a que el GSD agrega problemas y retos complejos al desarrollo de software. Con respecto a esto, la gestión de proyectos se puede definir como la implementación de un conjunto de herramientas y técnicas que permiten administrar el uso de diversos recursos para la realización de una tarea única y compleja, la cual está sujeta a limitaciones de tiempo, costo, y calidad, entre otros aspectos. Es así que el equipo asignado a la gestión del proyecto es responsable de los resultados obtenidos y, por lo tanto, debe asegurar que las metas y objetivos del proyecto sean alcanzados. Por consiguiente, la gestión eficaz y eficiente de los proyectos es un factor crítico para alcanzar el éxito.

Considerando lo anterior, el estudio de Monasor (2014) presentó una revisión de literatura para identificar estudios que tuvieran como objetivo el analizar o mejorar alguna de las diferentes fases del ciclo de vida del software en entornos de GSD. Como resultado de esta revisión se pudo determinar que los estudios publicados en la literatura se enfocan principalmente en las áreas de: gestión de los recursos humanos, gestión organizacional, infraestructura, y gestión de proyectos (véase Figura 2.3). Cabe destacar que la gestión de proyectos ha tomado gran relevancia en la literatura como factor importante para alcanzar el éxito de los proyectos de GSD, debido a que, en estos entornos globales la complejidad de los procesos organizacionales, la programación y asignación de tareas, y la estimación de costos del proyecto se tornan más complicadas debido a los problemas originados por las distancias temporal, geográfica, y de culturas.

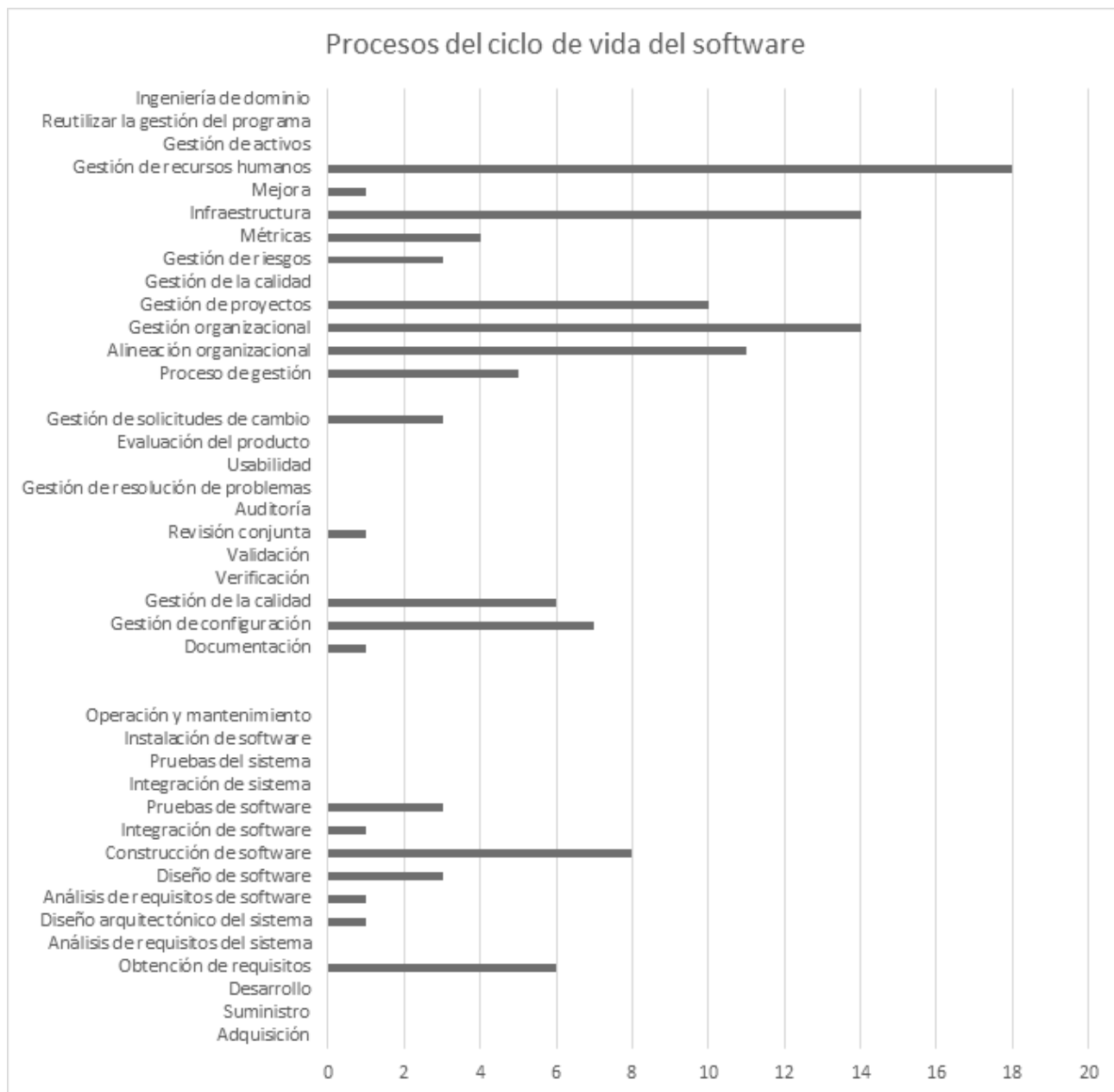


Figura 2.3. Trabajos relacionados en la mejora o análisis de los procesos software (Monasor, 2014)

Por otro lado, la gestión de los proyectos de software requiere de habilidades para la integración de tecnologías, economías, y relaciones humanas en el contexto del desarrollo de software. En consecuencia, la gestión no es una tarea sencilla, debido a que se debe interactuar con personas, equipos, y recursos organizacionales. Además, el GSD agrega aún más dificultades a las tareas de gestión, ya que los administradores del proyecto deben tratar ahora con equipos que se encuentran geográficamente distanciados, y cuyos integrantes pueden tener diferentes culturas y lenguas. De manera que las habilidades de gestión requeridas para entornos de GSD se convierten en factores importantes de éxito. En este contexto, el líder del proyecto es responsable de asegurar la visibilidad de las tareas, de supervisar al equipo de trabajo, de monitorear la estructura del proyecto, y de definir la estructura de los reportes. Sin embargo, es muy probable que dicho responsable no tenga contacto frente a frente con su equipo de trabajo, lo cual hace más difícil su tarea. Debido a lo anterior, los futuros ingenieros de software deben adquirir, a través de su formación universitaria, las capacidades y habilidades necesarias para desempeñar adecuadamente el rol de líder de proyectos en entornos de GSD. El estudiante debe desarrollar el conocimiento necesario para asignar roles y responsabilidades a los integrantes de un equipo virtual y debe asegurarse de que éstos puedan trabajar juntos de manera correcta, esto implica que debe poseer el conocimiento para identificar y gestionar las habilidades de cada integrante de su equipo de trabajo en beneficio del proyecto. De manera similar, el líder de proyectos debe mantener a su equipo motivado (Colomo-Palacios et al., 2014).

Es importante mencionar que diversos enfoques han sido propuestos para mejorar la enseñanza de estas habilidades, tal es el caso de Cajander, Clear, y Daniels (como se citó en Clear et al., 2015), que implementaron el concepto de “profesor externo”. Este concepto consistía en que un experto externo apoyara a los estudiantes en los problemas complejos de gestión y liderazgo. Una propuesta un tanto diferente es la de Daniels, Cajander, Pears, y Clear (como se citó en Clear et al., 2015) quienes implementaron los denominados “proyectos grupales abiertos” como una estrategia pedagógica para desarrollar las competencias profesionales de los estudiantes. Para lograr dicho objetivo, los estudiantes deben trabajar en un proyecto de colaboración global desafiante y auténtico, en el que deben asumir la responsabilidad de la dirección y gestión del proyecto y su propio aprendizaje dentro de el mismo. Es importante destacar que la enseñanza del GSD a través del desarrollo de cursos y proyectos entre universidades de diferentes lugares implica diversos desafíos como los mencionados anteriormente; no obstante, las universidades han notado la importancia de preparar a los estudiantes en ésta área y han implementado métodos de enseñanza que permiten a los estudiantes obtener experiencia práctica a través de proyectos de GSD, al mismo tiempo que aprenden la importancia de la habilidades suaves y del trabajo en equipo.

2.2. El enfoque pedagógico en la implementación del GSD

La enseñanza del GSD representa una combinación de estrategias de aprendizaje y enseñanza que tienen por objetivo inculcar en los estudiantes las habilidades y conocimientos necesarios para trabajar bajo este enfoque. Con respecto a esto, la literatura ha reportado diversos estudios relacionados con la educación este tópico, que se enfocan principalmente al fortalecimiento de habilidades suaves en los estudiantes. En este sentido, una cantidad importante de experiencias sugieren que la manera más efectiva de enseñar el GSD es haciendo uso de métodos de enseñanza activos, principalmente aquellos que siguen los principios del constructivismo como teoría de aprendizaje. De ahí que la manera tradicional de enseñanza basada en clases magistrales y lecturas, a través de las cuales el estudiante debe de memorizar los conceptos expuestos en los libros y por el profesor, no se considera un método de enseñanza efectivo en el contexto del GSD. No obstante, la

exposición de conceptos y el uso de lecturas pueden brindar a los estudiantes una idea general y teórica de los problemas que pueden afrontar en este enfoque, siempre y cuando estos conceptos sean aplicados en alguna práctica activa para que el estudiante pueda reforzar el aprendizaje de los conceptos teóricos (Ende et al., 2013).

Diversos enfoques pedagógicos han sido aplicados a la enseñanza del GSD. Por ejemplo, Braun, Dutoit, Harrer, y Brugge (2002) hicieron uso del enfoque “aprender haciendo” para intentar mejorar la enseñanza de este tema. Años más tarde, los estudios de Ende et al. (2013), Bass, McDermott, y Lalchandani (2015), y Ribaud y Saliou (2015) utilizaron el Aprendizaje Basado en Problemas (PBL, por sus siglas en inglés) para cumplir tal objetivo. De manera similar, los estudios de Gotel, Kulkarni, Scharff, y Neak (2008) y Fagerholm, Oza, y Münch (2013) hicieron uso del Aprendizaje Basado en Proyectos (ABP) con el mismo propósito. Por otro lado, Noll et al. (2014), Valencia, Vizcaíno, Garcia-Mundo, Piattini y Soto (2016), y Ghanbari, Similä, y Markkula (2015) implementaron el enfoque de juegos serios para apoyar la enseñanza del enfoque de GSD. Un enfoque diferente fue presentado por las investigaciones de Li et al. (2016) y Lappalainen et al. (2016) al hacer uso de las simulaciones como método de enseñanza. Por último, los esfuerzos de Bosnić, Žagar, y Crnković (2014) se enfocaron en implementar el concepto de *e-learning*.

Los enfoques pedagógicos citados anteriormente se derivan principalmente del constructivismo como teoría de aprendizaje. De acuerdo con Ribaud y Saliou (2015) el constructivismo tiene sus orígenes en la filosofía constructivista y se puede resumir mediante dos afirmaciones fundamentales:

- El aprendizaje se define como un proceso activo para la construcción de conocimiento más que como un proceso de adquisición de conocimiento.
- La enseñanza está esencialmente dirigida a ayudar a los estudiantes en este proceso, en lugar de transmitir conocimiento.

Hadjerrouit (2005) menciona que los enfoques basados en el constructivismo requieren de un conjunto de pautas y estrategias pedagógicas, tales como:

- **Construcción:** Los estudiantes son constructores de su propio conocimiento y no receptores pasivos de información. Es decir, los estudiantes deben construir nuevo conocimiento a partir del que ya poseen. Por otro lado, los profesores asumen principalmente el rol de colaboradores, guías y facilitadores de aprendizaje, no simples emisores de conocimiento.
- **Habilidades cognitivas:** Se debe proveer a los estudiantes de técnicas y métodos que les ayuden a desarrollar habilidades cognitivas de análisis y razonamiento, pensamiento analógico, reflexión y autoevaluación.
- **Tareas auténticas:** Las actividades de aprendizaje deben enfocarse en problemas realistas y motivadores. Por lo tanto, el conocimiento debe construirse en contextos reales y útiles.
- **Casos relacionados:** Los estudiantes deben nutrirse de las experiencias y casos relacionados de otros estudiantes con el objetivo de que puedan utilizar esta información para reforzar sus conocimientos.
- **Colaboración:** El aprendizaje debería tener lugar en un entorno que fomente la colaboración, la negociación, y la interacción social.

- Tecnología de la información: La tecnología bien diseñada e implementada puede facilitar el aprendizaje constructivista cuando proporciona a los estudiantes la información que realmente necesitan.

El enfoque del PBL se define como una forma de construir cursos educativos en donde se enseña a los estudiantes a través de problemas relacionados con la práctica profesional. El objetivo es que los problemas planteados a los estudiantes sirvan como estímulo para el desarrollo de sus habilidades prácticas (Ribaud y Saliou, 2015). Así pues, este método de enseñanza se basa en la teoría constructivista y se centra en el estudiante más que en el profesor. De esta manera, el PBL es introducido con el objetivo de apoyar al enfoque tradicional de enseñanza donde el profesor transmite la información y el estudiante es un receptor pasivo, mediante la definición de métodos activos enfocados en el estudiante para fomentar el desarrollo de las habilidades requeridas para la resolución de problemas reales. Finalmente, la hipótesis central de este método de enseñanza establece que los estudiantes deben aprender de manera más efectiva, cuando el proceso de aprendizaje se enfoca en la resolución de un problema concreto (Ende et al., 2013). De manera concreta el PBL se fundamenta en seis puntos principales:

1. El aprendizaje se enfoca en el estudiante.
2. El aprendizaje tiene lugar con pequeños grupos de estudiantes.
3. Un tutor o profesor está presente como facilitador o guía.
4. Los problemas son presentados al inicio del curso, antes de cualquier sesión de estudio.
5. Los problemas presentados se utilizan como herramientas para generar el conocimiento y las habilidades de resolución de problemas son necesarias para resolver problemas reales.
6. La información nueva se adquiere a través del aprendizaje previo, los objetivos y las tareas.

Por otro lado, el ABP permite a los estudiantes aprender a través de sus acciones y de la aplicación de sus ideas teóricas. Con este método de enseñanza se pretende que el estudiante realice actividades realistas, similares a las que los profesionales realizan en entornos reales de trabajo. Además, se basa en la teoría constructivista de aprendizaje enfocado en el estudiante, a través del cual los estudiantes generan su propio conocimiento mediante la aplicación de sus ideas en proyectos reales de trabajo. Es importante mencionar que la interacción social toma relevancia en el ABP como factor determinante en la generación de conocimiento, ya que los estudiantes obtienen mejores resultados de aprendizaje cuando comparten, utilizan y debaten ideas con sus compañeros. De manera general, Krajcik y Blumenfeld (2006) argumentan que el ABP es un enfoque para el diseño de entornos de aprendizaje que contempla cinco características principales:

1. Inicia con una pregunta que de manera general expresa un problema a resolver.
2. Los estudiantes exploran la pregunta inicial participando en dinámicas grupales que pretenden exponer ideas para la resolución del problema inicial. De esta manera, el estudiante aprende y aplica nuevas ideas relacionadas con la disciplina de estudio.
3. Los estudiantes, profesores y demás miembros participantes en el proceso de aprendizaje participan en actividades colaborativas con el objetivo de hallar una solución al problema planteado al inicio del curso.

4. Los estudiantes cuentan con herramientas de aprendizaje que los ayudan a participar en actividades que normalmente están más allá de sus posibilidades.
5. Los estudiantes desarrollan un conjunto de productos entregables relacionados con la solución del problema planteado inicialmente.

Por otra parte, los juegos serios hacen referencia a la implementación de algún videojuego cuyo objetivo se centra en la educación, más que en el mero entretenimiento. En este contexto, los juegos serios son vistos como una herramienta de aprendizaje debido a su capacidad para equilibrar el entretenimiento, la interactividad y la jugabilidad de los juegos tradicionales, pero enfocados en algún objetivo educativo en particular. Aunado a lo anterior, este enfoque se centra en el aprendizaje como un desafío difícil pero gratificante, con el objetivo de aumentar el compromiso y motivación de los jugadores por aprender. Así pues, éstos permiten simular entornos realistas en donde el estudiante debe aplicar su conocimiento con el objetivo de mejorarlo y obtener habilidades relacionadas con el área de estudio. De acuerdo con Valencia et al. (2016), el aprender simulando una experiencia real puede ser más efectivo para mejorar la comprensión de lo que se aprende, en comparación de lo que ocurre cuando se aprende leyendo o escuchando. Finalmente, Dale (como se citó en Valencia et al., 2016) afirma que los estudiantes pueden recordar el 90% de lo que aprenden a través de la aplicación de los conceptos en la simulación de entornos reales de los juegos serios.

Las simulaciones, por otro lado, hacen referencia a un método de aprendizaje que tiene por objetivo involucrar a los estudiantes en un entorno seguro que simule una experiencia real, en la cual los estudiantes deben aplicar su conocimiento y habilidades para resolver una problemática del mundo real que será el tema principal de dicha simulación. Así, los estudiantes enfrentan, en un entorno seguro, los problemas que afrontarán en el mundo real, los cuales podrán resolver aplicando sus conocimientos y aprendiendo de sus errores.

El método de *e-learning* hace referencia a la forma de enseñanza en la cual se utiliza la red como tecnología de distribución de la información. Al hablar de red se hace alusión tanto a la red abierta (internet) como a la cerrada (intranet). Los cursos de *e-learning* se definen de manera general como aquellos donde la mayor parte de la información, exámenes, y clases se proporcionan a través de recursos que están accesibles en la *web*, lo cual sugiere, que éstos estén disponibles en todo momento para el estudiante. En este sentido, el *e-learning* es relacionado generalmente con el concepto de “educación a distancia”, en donde se hace uso de las TI para brindar un aprendizaje interactivo, flexible, y accesible, a cualquier receptor potencial. De acuerdo con Cabero (2006) las principales características del *e-learning* son:

- El aprendizaje se da a través del uso de las TI.
- Se requiere del uso de navegadores *web* para acceder a la información.
- La interacción entre el profesor y el alumno es limitada y se da generalmente por medio de videoconferencias.
- Se utilizan diferentes herramientas de comunicación, tanto síncronas como asíncronas.
- Se hace uso de contenido multimedia (videos, audios, archivos digitales, etc.).
- La información y material del curso es almacenado, mantenido, y administrado en algún servidor *web*.
- El aprendizaje se considera flexible (el estudiante aprende a su propio ritmo).

- El aprendizaje es apoyado por tutorías.
- Se debe dar un aprendizaje individual más que colaborativo.
- El aprendizaje se desarrolla a través de un entorno altamente interactivo.

Finalmente, el método de enseñanza de “aprender haciendo” considera que la manera más efectiva de aprender algo es, precisamente, realizando o practicando lo que se desea aprender. Esta teoría de aprendizaje también se deriva del constructivismo y se enfoca en el estudiante. Así pues, tiene como principal objetivo fomentar el desarrollo de habilidades y el aprendizaje de información objetiva en el contexto de cómo puede ésta ser usada. Schank, Berman, y Macpherson (1999), mencionan algunos de los principios sobre los cuales se fundamenta dicha teoría son los siguientes:

- Aprender a hacer (habilidades), y no solo saber (conocimiento real).
- Aprendizaje en el contexto de un objetivo que es relevante, significativo, e interesante para el estudiante.
- Enseñanza de conocimiento por medio de tareas que están altamente relacionadas a cómo los estudiantes lo usarán fuera del entorno de aprendizaje.

En resumen, es evidente que la forma tradicional de enseñanza, en donde el profesor transmite a los estudiantes los conceptos teóricos y éstos deben de retener la mayor cantidad de información posible, no se adapta correctamente a la enseñanza actual de la IS en general y, concretamente, a la enseñanza del GSD. Por otro lado, las habilidades de los estudiantes que trabajarán bajo este enfoque serán determinantes para alcanzar el éxito de los proyectos. Así pues, los enfoques basados en la teoría constructivista, la cual promueve que los estudiantes sean constructores de su propio conocimiento y el profesor sea una guía en este proceso, son los que probablemente se adapten mejor a la enseñanza del GSD. Lo anterior, debido principalmente a que, como se ha mencionado anteriormente, las habilidades de los estudiantes son importantes para trabajar bajo el enfoque del GSD y estas habilidades pueden ser adquiridas por los estudiantes a través de la práctica en problemas reales, que en éste caso, pudiera ser a través de proyectos simulados, la resolución de problemas o la ejecución de actividades reales de trabajo (e.g., ABP, PBL, aprender haciendo), o entornos simulados en donde el estudiante aplique sus conocimientos (e.g., juegos serios, simulaciones). No obstante, gran parte de los trabajos realizados en el área de la enseñanza del GSD no toman en cuenta estos factores pedagógicos y realizan propuestas que, en su gran mayoría, no se sustentan en alguna teoría pedagógica de enseñanza. Por consiguiente, es importante proporcionar herramientas que ayuden en este proceso de enseñanza y que estén, al mismo tiempo, sustentadas por alguna teoría educativa o enfoque pedagógico.

2.2.1. Estado del arte

Como se ha mencionado anteriormente, cada vez más empresas de desarrollo de software se han percatado de los beneficios que representa el trabajar bajo entornos de GSD y, gracias al avance tecnológico, cada vez más de éstas deciden trabajar siguiendo este enfoque. No obstante, para poder obtener los beneficios del GSD, es necesario que las empresas se enfrenten a ciertos retos y complejidades que éste agrega al desarrollo de software, tales como: las distancias geográficas, la diferencia de horarios, la diferencia de lenguajes, entre otras. Es por esto que diversos investigadores y profesionales de la industria han sugerido soluciones a problemas concretos del GSD. Cabe destacar que una de las soluciones que se plantean en la literatura es proporcionar a los futuros

ingenieros “globales” de software una formación adecuada que les permita desarrollar las habilidades necesarias para trabajar bajo este enfoque.

La enseñanza del GSD presenta también una serie de problemas a resolver, dentro de los cuales se menciona que los métodos tradicionales de enseñanza resultan poco exitosos en la enseñanza de este enfoque en las aulas de clases. Dado este problema, en la literatura se sugiere utilizar métodos de enseñanza que permitan al estudiante aplicar sus conocimientos en un entorno real o simulado, en donde pueda aplicar de manera práctica los conceptos teóricos sobre el GSD. De esta manera, en las siguientes secciones se analizan algunas propuestas similares a la desarrollada en esta tesis, con el objetivo de presentar alternativas para la enseñanza del GSD, concretamente en el perfeccionamiento de las habilidades suaves requeridas para trabajar bajo este enfoque.

2.2.2. iBistro: Un entorno de aprendizaje para la construcción del conocimiento en los cursos de Ingeniería de Software de Sistemas Distribuidos

En los últimos años cada vez más empresas de desarrollo de software han optado por trabajar bajo un enfoque de GSD, debido principalmente a que la mezcla de personas con diferente conocimiento y cultura puede proveer a la organización de grandes beneficios. También gracias al avance tecnológico, las personas pueden colaborar desde diferentes lugares y se puede obtener mayor cercanía con clientes externos a la ubicación de la empresa. Sin embargo, los proyectos desarrollados siguiendo el enfoque del GSD introducen algunos problemas al ciclo de vida del software, debido principalmente a que los participantes de los equipos se encuentran distanciados geográficamente, y generalmente tienen diferentes culturas y lenguas. Como consecuencia de lo anterior, los proyectos pueden fracasar.

En este sentido, Braun et al. (2002) reportan sus experiencias en la enseñanza del GSD por medio de un curso entre la Universidad Técnica de Múnich (TUM, por sus siglas en inglés) en Alemania y la Universidad de Carnegie Mellon (CMU, por sus siglas en inglés) en Estados Unidos de América, el cual tiene como objetivo principal proveer a los estudiantes de experiencias realistas sobre el GSD.

2.2.2.1. Enfoque de enseñanza

El curso se basó en el trabajo en equipo entre los estudiantes de las dos universidades, quienes debían desarrollar un producto de software que se entregó a un cliente real. Durante el curso los estudiantes debieron colaborar a través de sitios *web*, correos electrónicos, repositorios, sistemas de control de versiones, entre otros, para diseñar y construir el producto. Por otro lado, las reuniones para que los estudiantes entregaran los avances del proyecto a los clientes y recibieran realimentación, se realizaron por medio de videoconferencias que facilitaron la comunicación. De acuerdo con los autores el desarrollo del curso tuvo buenos resultados, debido principalmente a que los estudiantes experimentaron el desarrollo de un proyecto de GSD, pudiendo entregar satisfactoriamente el proyecto solicitado y adquiriendo habilidades necesarias en este tipo de desarrollos. No obstante, se reportaron problemas de comunicación y colaboración entre los estudiantes durante el desarrollo del curso. Con relación a lo anterior, los principales problemas encontrados durante la ejecución del curso fueron:

- Dificultad para hallar *stakeholders* rápidamente: Los estudiantes experimentaron problemas tanto para encontrar a *stakeholders* que les pudieran proporcionar ayuda para el desarrollo, como para encontrar al responsable de algunas líneas de código, debido a que no se conocían entre ellos.

- Dificultad para acceder a la información del proyecto: Los estudiantes enfrentaron problemas para hallar la información del proyecto cuando ésta se documentaba durante una reunión informal o era tomada por un solo equipo.
- Dificultad para encontrar la última versión del software: Aunque los estudiantes trabajaron con un mismo software de control de versiones, la falta de comunicación y conocimiento de las herramientas hacía difícil las actualizaciones del código del sistema.

Los problemas mencionados anteriormente están altamente relacionados con las dificultades de comunicación entre los estudiantes. Al respecto, los autores clasificaron la comunicación en: comunicación formal y comunicación informal. Por un lado, la comunicación formal es generalmente no interactiva y hace referencia a especificaciones formales, documentación escrita, y reuniones estructuradas y programadas. Mientras que la comunicación informal generalmente ocurre en pares y es interactiva, generalmente conversaciones no programadas entre dos personas. En este sentido, los problemas presentes en el curso relacionados con la comunicación son concretamente ocasionados por la falta de comunicación informal. Debido a esto, se propuso un entorno de experimentación basado en el enfoque pedagógico de “aprender haciendo”, denominado iBistro, el cual permite a estudiantes y profesores capturar, organizar, y recuperar la información generada durante las conversaciones informales.

iBistro se definió como un espacio virtual en donde los estudiantes podían realizar reuniones informales; por lo tanto, se consideró como una fuente de adquisición de conocimiento informal. En consecuencia, iBistro capturó y estructuró automáticamente toda la información obtenida durante las reuniones en diferentes formas (e.g., identidad de los *stakeholders* participantes, actividad actual de los participantes, tema actual de la reunión, hora y ubicación, etc.). Aunado a lo anterior, los participantes podían colaborar para modificar el contenido de las minutas de cada reunión informal. Así mismo, esta plataforma hizo uso de las cámaras de las computadoras para observar la información del entorno físico desde donde se realizaba la reunión para cada equipo. En resumen, iBistro proporcionó un entorno donde se podía simular una sala de reunión informal con todos los elementos necesarios para que los estudiantes experimentaran una sensación de realidad, al mismo tiempo que intentó capturar la forma en que los estudiantes interactúan, con el objetivo de tener información del contexto en el que se toman las decisiones. Por otro lado, los estudiantes que no se encontraban físicamente con su equipo de trabajo, se podían integrar desde su estación de trabajo o asistente digital portátil mediante la interfaz de iBistro para acceder a la memoria contextual a través de una red.

Al mismo tiempo iBistro permitió que los participantes, que en un determinado momento no pudieron acudir a una reunión o llegaron tarde a ésta, pudieran tener conocimiento de la información tratada. Para lograr dicho objetivo, la plataforma registró, a través del audio y video, información relevante de cada reunión. Así también proporcionó un pizarrón electrónico en donde los participantes realizaron anotaciones, dibujos, gráficas, etc., que fueron almacenados como parte de la minuta de la reunión, la cual pudo ser consultada por cualquier miembro del equipo de trabajo que no estuviera presente en ésta, de tal modo que cualquier participante tomó ventaja de la cámara para tomar una captura de la información de sus compañeros. Así, los miembros del equipo accedieron a la información a través de una herramienta propia de iBistro denominada *MINUTEGENERATOR tool*.

iBistro fue diseñado siguiendo el modelo de sistemas de pizarrón (*blackboard systems*) propuesto por Nii (1986). Este modelo fue implementado en Inteligencia Artificial como una

alternativa para la resolución de problemas complejos no computables. Siguiendo las pautas de este modelo, iBistro se fundamentó en una arquitectura denominada “modelo de pizarrón concurrente distribuido”, como un enfoque para tratar la variedad de eventos, y su contexto, que ocurren durante las reuniones informales. De esta forma, el modelo permitió capturar eventos que ocurrieron durante una reunión, así como contenido de audio, video, y de pizarrón. Esta información capturada permitió que los integrantes del equipo tomaran decisiones acertadas, realizaran hipótesis, o propusieran una solución a un problema específico. En este sentido, la Figura 2.4 muestra el diagrama de clases que ejemplifica la arquitectura simplificada basada en el modelo de pizarrón de iBistro.

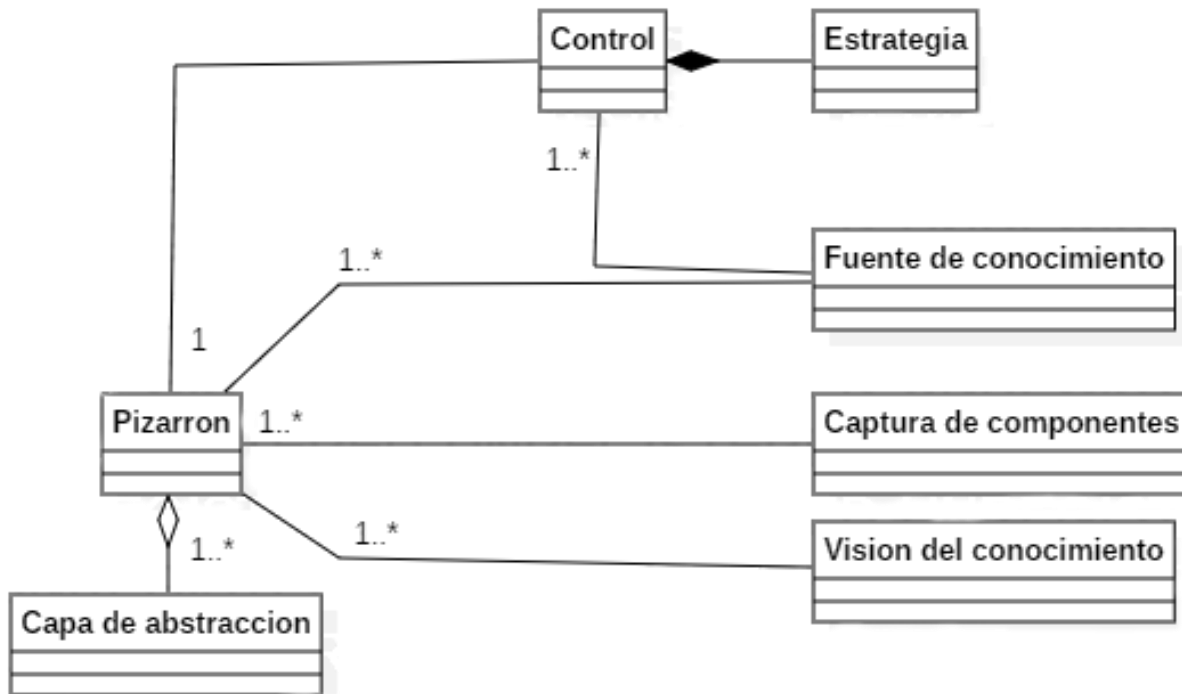


Figura 2.4. Arquitectura simplificada de iBistro (diagrama de clases obtenido de Braun et al., 2002)

Por otra parte, la adquisición y el almacenamiento de la información en iBistro se realizó considerando los eventos que ocurrían durante las reuniones y su ocurrencia en el tiempo. Así mismo, los usuarios pudieron registrar información en la base de datos de iBistro para actualizar el proyecto, en este caso, iBistro registró la fecha de creación y al usuario quien realizó las modificaciones. De manera similar, el control de versiones se registró en la base de datos de acuerdo con el usuario que realizó el cambio, y la fecha y ubicación del lugar donde se realizó la modificación. Por otro lado, la plataforma introdujo el concepto de editor de reuniones, que hizo referencia a una persona que se encargó de estructurar el conocimiento obtenido durante éstas. Dicha estructuración se realizó con base en los temas que se discutieron durante la reunión, el audio y video capturado, e información que esta persona consideró importante. El editor utilizó la herramienta proporcionada por iBistro para realizar su tarea (véase Figura 2.5). Finalmente, las minutas obtenidas por cada reunión fueron almacenadas, pudiendo incluir anotaciones externas. Por lo tanto, las minutas fueron almacenadas como archivos multimedia que los integrantes del equipo pudieron reproducir desde sus computadoras.

Para la realización de la plataforma los autores implementaron técnicas de los Sistemas Inteligentes de Tutoría (ITS, por sus siglas en inglés) y los Sistemas de Aprendizaje Colaborativo asistido por Computadora (CSCL, por sus siglas en inglés), con el objetivo de mejorar el entorno de aprendizaje y los procesos de trabajo. En este contexto, una de las técnicas implementadas es la denominada “modelos de usuario”, que permiten obtener información de cada usuario que trabaja con la plataforma. El modelo permite registrar el comportamiento de cada usuario ante diversos problemas. Esta técnica permite que los estudiantes obtengan información relevante acerca de sus compañeros y participantes en el proyecto y les permite decidir con mayor facilidad qué usuario o compañero podrá ayudarlos en determinado problema. Otra característica importante de iBistro es que, contiene información importante acerca de la IS que puede ser consultada por los estudiantes en cualquier momento. Esta información serviría de apoyo para los estudiantes cuando un profesor experto no está disponible para brindarle ayuda.



Figura 2.5. Herramienta para estructurar la información de una reunión (traducido de Braun et al., 2002)

2.2.2.2. Resultados y conclusiones

La implementación y pruebas de iBistro se llevaron a cabo con el desarrollo de un proyecto entre la TUM y la Universidad Nacional de Singapur. Durante las pruebas de la plataforma se identificaron problemas técnicos y deficiencias relacionadas con la calidad de audio y video, debido a la baja calidad del servicio de internet entre las dos universidades. No obstante, el modelo propuesto para la plataforma demostró ser adecuado para la captura y obtención del conocimiento y la navegación entre la información capturada. Así mismo, se validó la robustez de iBistro comparado con los medios tradicionales implementados para la comunicación informal (e.g., correo electrónico, mensajes instantáneos, entre otros).

2.2.3. Un aula virtual y global para enseñar Ingeniería Global de Software: Un proyecto de enseñanza de equipos mongol-alemanes

En la actualidad, es común que grandes proyectos de software sean divididos en una serie de componentes más pequeños, los cuales se asignan a diferentes equipos de trabajo, ubicados generalmente en diferentes partes del mundo. De ahí que el interés por el GSD haya crecido considerablemente durante la última década. No obstante, este enfoque enfrenta un conjunto de retos a superar como: la distancia geográfica, las diferencias de horarios, la confianza entre compañeros, las diferencias de culturas, y las diferencias de lenguajes. Por tal motivo, Ende et al. (2013) consideran que las habilidades técnicas de programación no son suficientes para que los ingenieros de software trabajen exitosamente bajo el enfoque del GSD, y sugieren que los profesionales de la industria adquieran, durante su formación profesional, un conjunto de habilidades suaves que les permitan trabajar bajo este enfoque, tales como:

- Dominio del idioma inglés.
- Habilidad para trabajar en equipos virtuales.
- Apreciación de la diversidad y multiculturalidad.
- Comprender la cultura y costumbres de otros países.
- Conocimiento de los protocolos de comunicación por medio de computadoras.
- Habilidades de comunicación.
- Capacidad de resolución de conflictos.
- Habilidades críticas y autocríticas.
- Capacidad de lidiar con la incertidumbre y la ambigüedad.
- Habilidades de trabajo en equipo.

2.2.3.1. Enfoque de enseñanza

El estudio de Ende et al. (2013), sugiere la aplicación en las aulas de clases del enfoque pedagógico de enseñanza del PBL en la enseñanza-aprendizaje del GSD, esto debido principalmente a que PBL deja de lado la forma tradicional de enseñanza centrada en el profesor y se enfoca en la enseñanza centrada en el estudiante. De acuerdo con este enfoque, el estudiante se convierte en el constructor de su propio conocimiento y desarrolla habilidades efectivas para la resolución de problemas en el área de estudio. En este contexto, el estudio reporta la ejecución en 2012 de un proyecto de enseñanza del GSD entre la Universidad de Ciencia y Tecnología de Mongolia (MUST) y el Instituto de Tecnología de Nuremberg (NIT) en Alemania. El objetivo principal de este trabajo era que los estudiantes adquirieran las habilidades necesarias para trabajar en proyectos de GSD. Para alcanzar tal objetivo, cuatro profesores del MUST viajaron a Nuremberg, Alemania para participar en un seminario de enseñanza, en donde se capacitó a los participantes del proyecto y se organizaron los equipos de trabajo.

Este proyecto contó con la participación de 20 estudiantes de maestría del NIT, quienes desempeñaron el rol de programadores, y cuatro profesores del MUST que desempeñaron el rol de cliente internacional o dueño del producto. El equipo de estudiantes debía obtener los requisitos del sistema y desarrollar un prototipo que sería entregado al dueño del producto. Cabe mencionar que el proyecto fue desarrollado siguiendo los principios de las metodologías ágiles de desarrollo, en este caso, los estudiantes siguieron las pautas de *Scrum*. La justificación de utilizar dicha metodología es

que ésta requiere que los programadores se encuentren en constante comunicación con los *stakeholders* y dueños del producto, con el objetivo de adaptar mejor el software a las necesidades del cliente.

El primer año de ejecución del proyecto fue exitoso, a tal grado que las universidades decidieron darle continuidad y agregar un nivel extra de complejidad al proyecto, la cual consistió en que los estudiantes tuvieran menos contacto directo con los *stakeholders* y se limitó la interacción entre éstos únicamente por medio de videoconferencias, mensajes instantáneos, y correo electrónico. Así, en 2013 participaron en el proyecto dos equipos de estudiantes distribuidos geográficamente de la siguiente manera: 8 estudiantes del MUST en Mongolia y 12 estudiantes del NIT en Alemania. Aunado a lo anterior, un profesor de cada universidad asumió el rol de profesor y asesor para cada equipo de trabajo. Así mismo, se impartieron clases intensivas durante una semana a los estudiantes participantes. Como resultado del ajuste del proyecto, se pudieron identificar dificultades de lenguaje y cultura debido a la falta de comunicación directa entre los estudiantes y los *stakeholders*, también se identificaron problemas de conexión a internet, lo cual agregó complejidad para el desarrollo del proyecto. De manera similar, la diferencia de horarios y la distancia geográfica agregaron complejidad al desarrollo. Por último, los autores reportaron que uno de los principales problemas para el proyecto fue la diferencia de culturas. En este contexto, los principales problemas relacionados con la diferencia de cultura fueron los siguientes:

- Distancia de poder: Los estudiantes de Mongolia suelen tener una relación altamente formal con sus profesores, lo cual dificulta la interacción entre éstos. Por otro lado, los estudiantes alemanes suelen tener una relación más relajada con sus profesores.
- Compañerismo e individualismo: Tanto los estudiantes alemanes como los estudiantes de Mongolia suelen tener mayor apego al trabajo individual sobre el colectivo.
- Falta de equidad de género: La mayoría de los estudiantes y profesionales de la industria del desarrollo de software son hombres, lo cual provoca que se promuevan ciertos comportamientos de superioridad sobre las mujeres.
- Incertidumbre y ambivalencia: Los estudiantes de Mongolia suelen no seguir las reglas, lo cual provoca conflictos en la toma de decisiones grupales. Por otro lado, la cultura alemana valora las reglas y la organización. Esta falta de entendimiento puede provocar ambigüedades e inseguridades.
- Orientación a largo plazo: El respeto por la tradición y el cumplimiento de las obligaciones sociales se valoran más que el ahorro y la perseverancia para ambas culturas.

Por otro lado, los problemas de conexión a internet provocaron retardos en la comunicación que afectaron directamente el trabajo de los equipos, debido a que el equipo de Mongolia presentaba graves problemas de conexión y, cuando podían conectarse, esta conexión presentaba diversos problemas de velocidad, por lo que establecer videoconferencias y llamadas de audio simultáneas resultaba una tarea difícil de llevar a cabo. Como resultado de lo anterior, los equipos optaron por establecer como medio de comunicación estándar los mensajes instantáneos. En este sentido, la comunicación escrita no verbal puede ser una ventaja cuando los miembros del equipo tienen dificultades para comunicarse en inglés, dado que este medio de comunicación permite a los estudiantes traducir los mensajes a su idioma natal. Adicionalmente, los estudiantes alemanes documentaron de manera escrita todo el progreso del proyecto y lo compartieron con los estudiantes

de Mongolia, debido a los problemas de conexión a internet, como resultado de esta actividad, la cohesión del equipo mejoró.

Por otra parte, la diferencia horaria entre los dos países participantes provocó que el tiempo para la comunicación directa se redujera a solamente una hora. Por tal motivo, la planificación y asignación de actividades fueron necesarias, lo cual requirió un esfuerzo extra de los estudiantes para relacionarse con tópicos sobre la planificación.

2.2.3.2. Resultados y conclusiones

Los estudiantes enfrentaron diversos problemas durante el proyecto, debido principalmente a las diferencias de cultura, diferencia de horarios, diferencia de lenguajes, y distancia geográfica. Así mismo, la mayoría de los participantes no habían trabajado antes en algún proyecto global, por lo cual presentaron un alto grado de resistencia al cambio en su manera de trabajar. No obstante, los estudiantes estaban motivados por obtener buenas calificaciones y se enfocaron en la entrega del producto final y no en el proceso de desarrollo global. Por tal motivo, cuando entregaron el producto se sintieron decepcionados, ya que consideraban que podían haber realizado un mejor trabajo. Sin embargo, el estudio reporta que los estudiantes mejoraron sus habilidades de resolución de problemas, ya que durante el proceso de desarrollo tuvieron que improvisar sus propias soluciones bajo presión de tiempo, al mismo tiempo que pudieron enfrentar de manera real los problemas que se presentan en el GSD.

2.2.4. Una plataforma para la enseñanza del desarrollo de software distribuido

La enseñanza del GSD puede realizarse de diferentes maneras, la más conocida es la clásica forma de enseñanza que se desarrolla en las aulas de clase y que generalmente se limita a la transferencia de información y conocimiento sobre los métodos y las técnicas relacionadas con algún tema en particular, pudiendo realizar ejercicios que simulen el tema de estudio en un entorno real. Generalmente, los ejercicios realizados son pequeños y poco realistas, por lo tanto, los estudiantes no experimentan realmente los problemas y retos del área de estudio. Con respecto a la enseñanza del GSD, este método de enseñanza resulta ser poco útil, debido a que no se puede mostrar a los estudiantes la complejidad, problemas y retos presentes en este entorno de trabajo. Por otro lado, en un entorno real del GSD los estudiantes pueden experimentar los efectos de los métodos y técnicas de la IS y su relevancia en la forma en que aplican su conocimiento. Diversos estudios han demostrado que enseñar a los estudiantes por medio de experiencias prácticas, ayuda a mejorar el rendimiento tanto individual como de equipo con relación al GSD. Otra alternativa sugerida en la literatura son las simulaciones que permiten que los estudiantes experimenten y aprendan en un entorno seguro.

2.2.4.1. Enfoque de enseñanza

En la literatura se han reportado diversos estudios que se enfocan en la enseñanza práctica del GSD, tal es el caso de Fagerholm et al. (2013) que proponen un curso a nivel de maestría en el Departamento de Ciencias de la Computación de la Universidad de Helsinki, para el desarrollo de un proyecto bajo el enfoque del GSD. El estudio hace uso del “Laboratorio de Fábrica de Software” de la misma universidad. Esta fábrica de software es un laboratorio que provee a estudiantes y profesores de un entorno para la investigación y enseñanza de la IS. Este laboratorio ha sido utilizado como una plataforma para la enseñanza de la IS en estrecha colaboración con la industria. El objetivo es proveer a los estudiantes de un entorno realista que les permita integrar los conocimientos previos de informática e IS con conocimientos prácticos en la realización de proyectos de software reales. Así pues, los estudiantes participantes en el curso propuesto deben

trabajar en dicho laboratorio. Para ello, los estudiantes trabajan un promedio de seis horas por día en el laboratorio durante cuatro o cinco días a la semana. Los proyectos tienen una duración aproximada de siete semanas. Durante este tiempo los estudiantes aplican técnicas y métodos de las metodologías ágiles de desarrollo para producir rápidamente un prototipo funcional de software, esto en colaboración constante con un cliente externo. Por otro lado, el laboratorio permite que los proyectos se lleven a cabo de una manera que simula la realidad del desarrollo de software en la industria de la manera más real posible. Entonces, los estudiantes desarrollan los proyectos en un entorno que simula una pequeña empresa de desarrollo, en la cual, pueden desarrollar nuevos proyectos de software o bien pueden desarrollar mejoras para algún proyecto existente. En este contexto, los proyectos se desarrollan en un entorno de laboratorio estandarizado, pero personalizable con un diseño físico específico (mobiliario, equipamiento tecnológico y diferentes zonas presentes en una empresa de desarrollo), infraestructura técnica bien definida e infraestructura para la realización de estudios empíricos.

El laboratorio provee a los estudiantes de experiencias realistas que les permiten integrar el conocimiento teórico y práctico del GSD, permitiéndoles al mismo tiempo desarrollar habilidades duras y suaves para su posterior integración al mercado laboral. También se espera que los estudiantes asuman la responsabilidad del desarrollo de todo el proyecto, esto incluye que asuman roles y responsabilidades relacionadas con la gestión del proyecto, comunicación con los clientes, gestión de requisitos, mejora continua del proceso de desarrollo y, el desarrollo del software en sí. Aunado a lo anterior, los profesores pueden complementar la enseñanza con otros cursos que consideren necesarios (e.g., liderazgo de proyectos de software, gestión de proyectos, dinámica de grupos, arquitectura de software y procesos de software), así como cursos intensivos sobre temas técnicos (e.g., control de versiones, lenguajes de programación y pruebas).

Finalmente, el proceso de enseñanza puede ser dividido en tres fases principales: pre-proyecto, desarrollo del proyecto, y post-proyecto, los cuales se describen a continuación:

- La fase de pre-proyecto tiene como objetivo llevar el proyecto a un estado en el cual pueda ser entregado al equipo de programadores para su implementación. De manera general, los pasos a seguir durante esta fase son:
 - Proyección y selección de proyectos: Durante esta fase se hace la selección de una idea del proyecto a desarrollar y se realiza la proyección y alcance del mismo a través de diferentes propuestas e ideas del equipo de trabajo.
 - Inscripción: Se realiza una fase de inscripción en donde se selecciona a los estudiantes que participarán en el desarrollo del proyecto.
 - Identificación de problemas administrativos: Una vez que el equipo de trabajo se haya conformado, pueden surgir diversos problemas administrativos (cuentas de usuario, horarios de trabajo, acceso a la infraestructura, etc.) que deben ser tratados.
 - Inicio del proyecto: Durante esta fase se introduce a los estudiantes al problema y objetivos del proyecto, al mismo tiempo que se capacita al cliente y estudiantes para mejorar la comunicación entre ellos.
- Durante la fase de desarrollo, se realizan reuniones semanales en donde el grupo de estudiantes entrega avances y muestra el estado del proyecto al cliente quien dará realimentación del trabajo realizado. Las reuniones se realizan por medio de tecnologías de comunicación que permiten al cliente tomar el control de la reunión. Aunado a lo anterior, el

equipo de estudiantes realiza reuniones constantes siguiendo los lineamientos establecidos en *Scrum*.

- Por último, en la fase del post-proyecto se realizan actividades de evaluación a los participantes del proyecto con el objetivo de evaluar sus experiencias durante el desarrollo.

2.2.4.2. Resultados y conclusiones

El uso de la fábrica de software para el desarrollo de los proyectos de GSD permitió reducir la carga administrativa a los profesores que pudieron enfocarse en los aspectos pedagógicos de la enseñanza del GSD. Por otro lado, el laboratorio permitió a estudiantes y profesores tener un contacto óptimo con los *stakeholders* del proyecto, debido al soporte de diversas herramientas y tecnologías con las que éste dispone. Así mismo, estudiantes, profesores e investigadores pudieron comprender de mejor manera los problemas y retos en la distribución de trabajo que sugiere el GSD.

2.2.5. GSDGame: Un juego serio para adquirir las competencias necesarias en el GSD

La globalización ha permitido que cada vez más empresas de desarrollo de software produzcan bajo entornos de GSD, debido principalmente a los beneficios que este enfoque presenta. Sin embargo, el GSD presenta aún una serie de problemas que dificultan su uso, tales como la deslocalización de equipos que agrega problemas de comunicación, coordinación y control, y problemas originados de la unión de diferentes culturas. Las diferencias de lenguaje afectan directamente a la comunicación y confianza entre compañeros. De tal modo que es importante que las personas que trabajan en el GSD posean las capacidades y habilidades necesarias requeridas por este tipo de entornos de trabajo y esta formación debe iniciar desde la preparación universitaria de los futuros ingenieros “globales” de software. No obstante, la enseñanza de estas habilidades no es una tarea sencilla de realizar, principalmente por que la forma tradicional de enseñanza basada en lecturas no se adapta correctamente a la enseñanza del GSD.

2.2.5.1. Enfoque de enseñanza

Dados los problemas mencionados anteriormente, Valencia et al. (2016) proponen la implementación de un juego serio denominado GSDGame para la enseñanza de las competencias necesarias en el GSD. Los autores afirman que actualmente la forma de enseñanza se centra en que el estudiante aprenda las habilidades necesarias para su desarrollo profesional, más que en la forma en cómo el profesor enseña. En este sentido, los juegos serios permiten a los estudiantes desarrollar habilidades y competencias necesarias para su vida profesional en un entorno seguro. La principal característica de éstos es su capacidad para balancear el entretenimiento, la interactividad, y la jugabilidad de los juegos típicos, pero con un objetivo educativo al mismo tiempo que mantienen la motivación de los estudiantes en el proceso de aprendizaje.

Con esta idea en mente, GSDGame permite que los estudiantes asuman el rol de líder de proyectos de un equipo virtual que debe desarrollar un producto de software, mientras que se enfrenta a los problemas que el GSD plantea. El juego se compone de un conjunto de escenarios que simulan diferentes eventos y problemas presentes en el GSD. Cada escenario se compone de un nombre, una duración, un presupuesto, los módulos de componentes, y los países involucrados. Las principales características del juego son:

- El juego simula un conjunto de eventos y problemas inesperados que pueden ocurrir en los proyectos de GSD.

- El juego se compone de diferentes escenarios, los cuales tienen variedad de niveles y dificultad que incrementa a medida que el usuario avanza en el progreso del juego.
- El juego tiene la capacidad de simular conversaciones en línea, correo electrónico y llamadas telefónicas, para que el estudiante experimente la comunicación asíncrona presente en el GSD.
- Cada vez que ocurra un evento o problema inesperado, el estudiante tiene la posibilidad de seleccionar de una lista preestablecida de soluciones la que mejor se adapte para resolver el problema en cuestión. Dependiendo de qué tan bien resuelva el problema, la elección del estudiante se corresponde con la cantidad de puntos obtenidos.
- El juego cuenta con un sistema de puntos que se actualiza con relación a las decisiones tomadas por el estudiante cuando un problema ocurre, y dependiendo del número de días y presupuesto restantes para la entrega del producto.
- El estudiante puede solicitar ayuda en cualquier escenario del juego.
- El juego permite al estudiante modificar su información y ver las instrucciones del juego.
- El estudiante puede interactuar con los empleados virtuales con los que trabaja en el juego.

El juego está compuesto de dos subsistemas principales, uno para estudiantes y otro para profesores, quienes son los encargados de proveer los escenarios y de supervisar las sesiones de juego. Tanto los estudiantes como los profesores deben estar previamente registrados en el juego para poder interactuar con éste. Cuando un estudiante inicia el juego podrá ver la información del proyecto y deberá configurarlo, así mismo podrá iniciar en cualquier momento una comunicación asíncrona con su equipo o ver el calendario de trabajo (véase Figura 2.6).



Figura 2.6. Interfaz de estudiante para GSDGame (Valencia et al., 2016)

Por otro lado, el profesor tiene la capacidad de agregar problemas al juego (véase Figura 2.7), realizar llamadas de voz, iniciar conversaciones, crear proyectos (véase Figura 2.8), ver los resultados de cada estudiante, entre otras acciones.

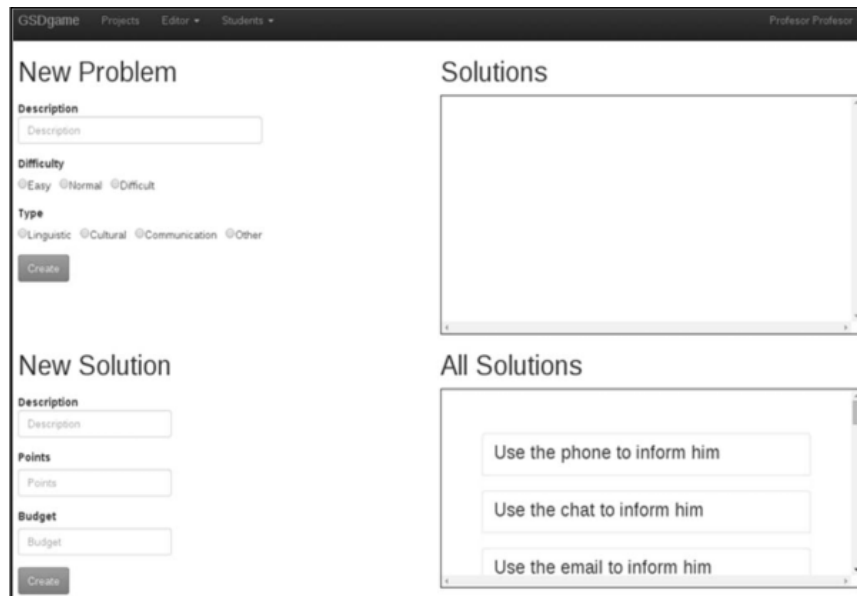


Figura 2.7. Interfaz para agregar problemas al juego (Valencia et al., 2016)

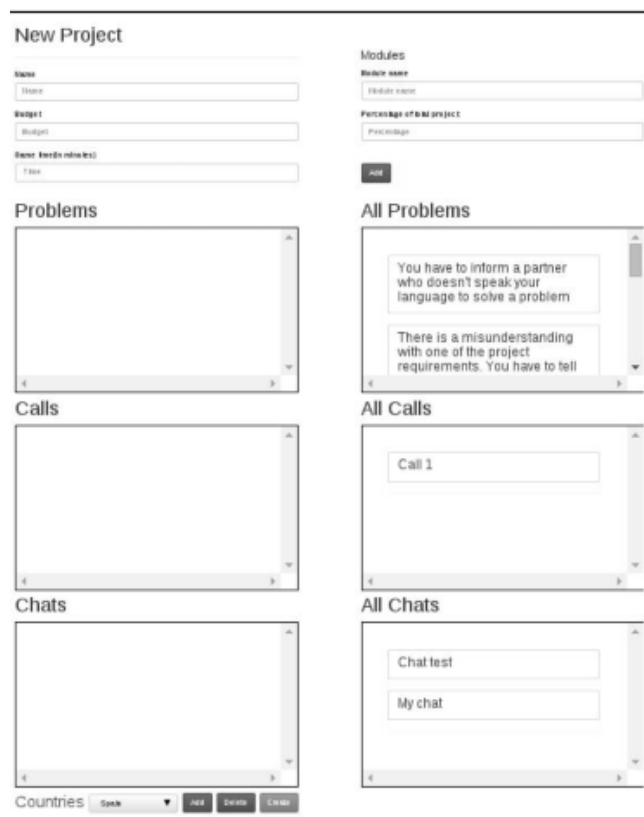


Figura 2.8. Interfaz para agregar un nuevo proyecto al juego (Valencia et al., 2016)

2.2.5.2. Resultados y conclusiones

Para poder medir la eficacia del juego propuesto, fue necesaria la evaluación de un experto en juegos serios. Dicha evaluación consistió en verificar cada una de las funcionalidades implementadas en GSDGame. Es importante destacar que el juego no se evaluó en un caso de estudio, sino que la evaluación tomó en cuenta los aspectos de funcionalidad de éste con relación a las características de los juegos serios y su relación con el objetivo pedagógico con el que fue diseñado, en este caso, la enseñanza del GSD.

Como resultado de la evaluación, los autores obtuvieron los siguientes resultados:

- Alcance de los objetivos: Todas las funcionalidades del juego se relacionaron directamente con los objetivos educativos.
- Mecánica del juego: La mecánica del juego se relacionó en gran medida con los objetivos y proporcionó a los estudiantes diversos desafíos y retos que debieron cumplir para obtener alguna recompensa.
- Progreso del juego: El juego permitió que el estudiante controlara su progreso durante la sesión de juego, esto se observó principalmente desde el inicio del juego.
- Recompensas y premios: El juego otorgó al estudiante puntos que éste puede acumular, no obstante, no fue clara la forma en cómo pueden ser obtenidos estos puntos.
- Compartir desafíos y logros: El juego no permitió que los estudiantes compartieran con sus compañeros los logros o desafíos cubiertos.

En resumen, el juego propuesto cumplió efectivamente con los objetivos educativos para los cuales fue diseñado; no obstante, se considera de baja calidad con relación a las medidas de personalización, realismo, argumento y narración de la historia, y adecuación de la dificultad en los niveles del juego. Finalmente, los autores concluyen en que es necesario mejorar el juego con relación a las recomendaciones hechas por el experto. Además, es importante realizar un caso de estudio que permita verificar la efectividad del juego en un entorno académico.

2.2.6. Enseñando un curso de desarrollo global de software: Experiencias de los estudiantes utilizando la simulación de ejercicios en el aula

En la actualidad diversas empresas de desarrollo de software se han enfocado a implementar el GSD con el objetivo de obtener los beneficios que este enfoque de trabajo trae consigo. Por tal motivo, es importante ofrecer una correcta enseñanza en esta área a los estudiantes de la IS. Sin embargo, la forma tradicional de enseñanza en las aulas universitarias no cumple con los requisitos del GSD, debido a que la enseñanza de este enfoque sugiere preparar a los estudiantes en nuevas teorías, herramientas, y habilidades.

2.2.6.1. Enfoque de enseñanza

Debido a lo expuesto anteriormente, las universidades han optado por establecer cursos de GSD a nivel mundial, en los cuales participan estudiantes de diferentes universidades y países con el objetivo de desarrollar un proyecto de software. Pero, el llevar a cabo estos cursos muchas veces requiere de un mayor esfuerzo, tanto administrativo como educativo, por parte de profesores y universidades. Por tal motivo, Lappalainen et al. (2016) llevaron a cabo un curso de GSD a nivel maestría para estudiantes de la IS. Dicho curso tuvo como objetivo estudiar los retos y soluciones del GSD siguiendo dos enfoques principales: por un lado, se impartieron clases teóricas a los

estudiantes acerca de los conceptos y técnicas del GSD y, por otro, los estudiantes realizaron ejercicios de simulación relacionados con el desarrollo de algún proyecto real desarrollado con el enfoque del GSD. Las clases impartidas a los estudiantes se enfocaron en la distribución del trabajo en los diferentes sitios del desarrollo, en los problemas de comunicación, las diferencias de horarios, y los problemas multiculturales. En total, se dedicaron 70 horas a estas actividades con el objetivo de que los estudiantes pudieran conocer los factores de éxito del GSD, así como conocer las herramientas y métodos que pudieran aplicar en este enfoque. Por otro lado, la simulación se enfocó en el desarrollo de un proyecto de GSD por un equipo virtual, haciendo uso de los métodos y herramientas apropiados durante aproximadamente de 4 a 5 semanas. Los ejercicios consistieron en una introducción al inicio del curso para mostrar a los estudiantes los objetivos de los ejercicios y el dominio del problema en el que trabajarían, y una sesión de discusión al final de éste. Los estudiantes conformaron equipos de 3 a 5 estudiantes que participaban en algún proyecto junto con otros equipos.

La simulación permitió a los estudiantes experimentar los problemas presentes en el GSD a través de la simulación de sitios ubicados geográfica y temporalmente distintos, en los cuales tuvieron que ser distribuidos los equipos participantes en el desarrollo del proyecto. Los roles que los estudiantes desempeñaron durante la ejecución del curso fueron: un equipo de desarrolladores, un equipo de clientes que solicitaban los requisitos del proyecto, y un equipo de usuarios que utilizarían el software. El equipo de clientes debería de solicitar el desarrollo de algún proyecto al equipo de desarrolladores. En un inicio, los clientes deberían proveer los requisitos iniciales del proyecto, una vez que se obtuvieran los requisitos, el equipo de desarrolladores debería de construir el software para que posteriormente el equipo de usuarios validara lo entregado con las necesidades del negocio. Los estudiantes siguieron los principios de las metodologías ágiles para realizar el proyecto.

Durante la ejecución del curso los profesores obtuvieron información de los estudiantes participantes en forma de reportes. En cada reporte los estudiantes analizaron su aprendizaje y la posibilidad que tenían de aplicar lo aprendido en un entorno real de trabajo, estos reportes fueron parte del trabajo que los estudiantes debían realizar como actividades del curso, además podían dar realimentación acerca de éste en sus reportes.

2.2.6.2. Resultados y conclusiones

Con el objetivo de determinar el aumento en el conocimiento de los conceptos del GSD en los estudiantes los autores realizaron un análisis de los reportes documentados. Cada reporte fue clasificado con una palabra clave que representaba un problema en específico con el que se enfrentaron los estudiantes, esto permitió determinar los mayores problemas a los que se enfrentaron durante el desarrollo de los proyectos de GSD. Por otro lado, se aplicó una encuesta de evaluación del curso a 22 estudiantes con el fin de obtener y analizar las opiniones de éstos con respecto al curso. Los resultados obtenidos de la aplicación de la encuesta mostraron que más del 50% estuvieron totalmente de acuerdo en que las lecturas y clases teóricas les ayudaron a entender y prepararse de mejor manera en el enfoque GSD. De manera similar, cerca del 95% de los estudiantes coincidieron en que las clases teóricas les ayudaron a formar su propio conocimiento. Por otro lado, el 82% estuvo de acuerdo en que los ejercicios simulados les sirvieron de ayuda para comprender los aspectos relevantes del GSD. Finalmente, el promedio de calificaciones finales de los estudiantes fue de 3.85 de 5 alcanzables, lo cual demuestra que pudieron aplicar correctamente los conceptos teóricos en los ejercicios simulados.

2.2.7. Consideraciones finales sobre el estado del arte

En los últimos años, diversas empresas se han unido al GSD debido principalmente a los beneficios que este tipo de desarrollo pretende otorgar a las empresas, tales como: el abaratamiento de costos de desarrollo, la obtención de jornadas de trabajo más extensas, la unión de diferentes culturas e ideas, entre otras. No obstante, obtener estos beneficios no es una tarea sencilla y, por el contrario, requiere de un mayor esfuerzo por parte de empresas y trabajadores para poder alcanzar los beneficios señalados anteriormente. En este contexto, investigadores y profesionales de la industria han sugerido un conjunto de posibles soluciones que ayudarían a obtener dichos beneficios. Es así que en la literatura se ha sugerido mejorar la enseñanza del GSD dentro de las aulas universitarias como una alternativa que ayudaría a los futuros ingenieros globales de software en su capacitación y mejoramiento de las habilidades requeridas por este tipo de desarrollo.

Mejorar la enseñanza sobre el GSD ha tomado relevancia en los últimos años como una de las posibles soluciones a los problemas y retos presentes en este enfoque. Así pues, en la literatura se reportan diversas propuestas que se han tomado en cuenta para la enseñanza del GSD en las aulas universitarias, tal es el caso de las simulaciones, los cursos entre universidades, los juegos serios y GBL, entre otros. Por tal motivo, en este capítulo se expusieron algunos trabajos relacionados con el proyecto presentado en esta tesis, que como se ha mencionado anteriormente, considera el desarrollo de un plataforma que permita el diseño e implementación de un curso de GSD entre universidades, tomando como base un marco de referencia que exponga las bases teóricas y pedagógicas que permitan establecer las pautas y lineamientos para la realización de un curso sobre el GSD a nivel licenciatura, también propuesto en este proyecto.

El problema de la enseñanza del GSD en las aulas universitarias sugiere que la forma tradicional de enseñanza, basada en clases magistrales, no se adapta correctamente a la capacitación de este enfoque, por tal motivo, las propuestas expuestas anteriormente sugieren utilizar métodos y técnicas que permitan a los estudiantes aplicar su conocimiento teórico acerca del GSD en un entorno, real o simulado, donde puedan desarrollar un proyecto. En este contexto, uno de los principales enfoques seguidos para cumplir este objetivo es la ejecución de cursos entre universidades de diferentes países, llevando a cabo el desarrollo de algún proyecto de software. Este enfoque pudiera ser considerado como la mejor manera de enseñar el GSD, dado que permite a los estudiantes experimentar de manera real los problemas y complejidades de este enfoque de trabajo. No obstante, resulta complejo poder llevar a cabo un curso entre universidades que tienen diferentes formas de trabajo, diferentes formas de evaluación, diferente cultura, horarios, etc. Además, la organización del curso requiere de un mayor esfuerzo por parte de las universidades y profesores para coordinarlo, y por parte de los estudiantes para trabajar con personas desconocidas. Por otro lado, los juegos y simulaciones permiten que los estudiantes experimenten la forma de trabajo del GSD en un entorno controlado que si bien disminuye los problemas que presenta la ejecución de un curso entre universidades, agrega nuevas complejidades como obtener la atención de los estudiantes y mantenerlos motivados.

Por otra parte, los trabajos expuestos anteriormente sugieren la enseñanza del GSD por medio del enfoque constructivista de enseñanza, que sugiere que los estudiantes sean constructores de su propio conocimiento. De manera similar, la propuesta presentada en esta tesis sugiere la enseñanza por medio de este enfoque y concretamente del método ABP. Cabe resaltar que, aunque los trabajos expuestos sugieren la aplicación de dicho enfoque de enseñanza, no se profundiza en los aspectos pedagógicos que sustenten los contenidos, métodos y herramientas aplicados en el proceso de enseñanza. Por tal motivo, la propuesta presentada en esta tesis contempla los aspectos pedagógicos para dar sustento a los contenidos y lineamientos que son presentados como parte del marco de

referencia que sustenta el desarrollo de la plataforma para la implementación de un curso de GSD. En este sentido, la propuesta también tomará en consideración las experiencias reportadas acerca de la ejecución de los cursos de GSD de los trabajos expuestos anteriormente. Por otra parte, la comunicación informal es tomada como el mecanismo de interacción que forma parte de la plataforma que es desarrollada; así mismo, se implementa el concepto de CWS añadiendo algunas modificaciones que permitan al líder de proyectos monitorear la comunicación de su equipo.

Finalmente, el estado del arte desarrollado ha permitido comprobar que la enseñanza del GSD en las aulas universitarias es una tarea compleja y que es importante ofrecer alternativas de enseñanza que permitan a los estudiantes desarrollar las habilidades y competencias requeridas por este enfoque de trabajo. De igual manera, se ha podido observar que se han obtenido buenos resultados y se han encontrado diversos problemas durante la enseñanza que pudieran servir como base para este proyecto con el fin de ofrecer alguna alternativa que pudiera obtener mejores beneficios.

A modo de resumen, la Tabla 7 resume una comparación de los trabajos expuestos en esta sección.

Tabla 7. Cuadro comparativo de investigaciones enfocadas en mejorar la enseñanza del GSD

Trabajo	Tipo de propuesta	Enfoque pedagógico	Objetivo	Características principales
iBistro (Braun et al., 2002)	Herramienta de comunicación	Aprender haciendo	Proveer a los estudiantes de experiencias realistas sobre el GSD	<p>Simular una sala de reuniones.</p> <p>Capturar y estructurar la información de las reuniones.</p> <p>Ver y modificar el contenido de las minutas.</p>
Una aula virtual y global (Ende et al., 2013)	Curso internacional de GSD	PBL	Proveer a los estudiantes de las habilidades necesarias para trabajar en proyectos de GSD	<p>Los profesores deben asumir el rol de clientes internacionales.</p> <p>Los estudiantes deben obtener los requisitos del sistema y crear un prototipo funcional de éste.</p> <p>Los estudiantes aplican técnicas de las metodologías ágiles.</p>
Una plataforma para la enseñanza del desarrollo de software distribuido (Fagerholm et al., 2013)	Curso internacional a nivel maestría de GSD	No especificado	Proveer a los estudiantes de un entorno realista que les permita integrar sus conocimientos en un entorno práctico	<p>Los estudiantes trabajan en un laboratorio que simula una empresa de software.</p> <p>Los estudiantes trabajan un promedio de seis horas durante cuatro o cinco días.</p> <p>Los proyectos tienen una duración de siete semanas aproximadamente.</p> <p>Los estudiantes aplican técnicas de las metodologías ágiles.</p>
GSDGame (Valencia)	Juego serio	No	Fomentar en los	Los estudiantes asumen el

Trabajo	Tipo de propuesta	Enfoque pedagógico	Objetivo	Características principales
et al., 2016)		especificado	estudiantes las competencias necesarias sobre el GSD	<p>rol de líder de proyectos.</p> <p>Los escenarios del juego simulan diferentes eventos y problemas presentes en el GSD.</p> <p>Consta de un subsistema para profesores quienes administran y gestionan el juego.</p> <p>Consta de un subsistema de interacción entre los estudiantes.</p>
Curso de desarrollo global de software (Lappalainen et al., 2016)	Curso teórico-práctico de GSD	Clases teóricas y simulaciones	Estudiar los retos y soluciones del GSD	<p>Se impartan clases teóricas sobre el GSD.</p> <p>Los estudiantes realizan ejercicios simulados de GSD.</p> <p>Se dedicaron un total de 70 horas en clases teóricas</p> <p>Los ejercicios simulados tuvieron una duración de cuatro a cinco semanas aproximadamente.</p> <p>Los estudiantes formaron equipos de tres a cinco integrantes para realizar los ejercicios.</p>

3. Desarrollo de la solución

Esta tesis es desarrollada considerando dos principales aportaciones: por un lado, un marco de referencia que establezca las pautas y lineamientos para el diseño y ejecución de cursos sobre GSD entre dos universidades, siguiendo los principios del ABP y, por otro lado, el diseño y construcción de una plataforma computacional que permita llevar a la práctica los principios teórico-prácticos establecidos por el marco de referencia. Por tal motivo, en las siguientes secciones se describirán los enfoques metodológicos seguidos tanto para el diseño del marco de referencia como para el diseño y construcción de la plataforma computacional, mencionados anteriormente.

3.1. Diseño del marco de referencia

Para el diseño del marco de referencia propuesto en esta tesis se seguirán los lineamientos establecidos en la metodología propuesta por O'Neill (2015) para el diseño curricular en la educación superior. Dicha metodología sugiere analizar primeramente el entorno para el que se pretende desarrollar el curso, debido a que el plan de estudios de un programa educativo se ve altamente influenciado por el entorno social, económico, físico, y cultural en el que se aplicará. Así pues, el estudio de Stark (como se citó en O'Neill, 2015) identificó nueve puntos clave que influyen en el diseño de los planes de estudio:

- Las características y metas de los estudiantes.
- Influencias externas.
- Metas del programa de estudio.
- Metas de las instituciones educativas.
- Factores pragmáticos.
- Literatura pedagógica.
- Asesoramiento disponible en las instituciones.
- Disponibilidad de instalaciones

Considerando lo anterior, esta metodología sugiere tres entornos principales que afectan directamente el diseño de los planes de estudio y que se consideran importantes de analizar para el diseño del marco de referencia propuesto:

- Contexto internacional: En el entorno internacional participan estudiantes de diversos países, por lo tanto, cada estudiante tiene diferente forma de trabajo y cultura, debido a esto se debe

diseñar el plan de estudios considerando las necesidades y características de una población estudiantil diversa. Además, se sugiere preparar a los estudiantes para desempeñarse en un mundo complejo y cambiante.

- Contexto nacional: En el entorno nacional se debe tomar en cuenta las diferentes regulaciones y formas de trabajo educativas del país en donde se pretende aplicar el curso.
- Contexto institucional: Cada institución tiene sus propios contextos sociales, históricos, económicos y políticos. Éstos impactan directamente en cómo el plan de estudios se desarrolla.

Teniendo en cuenta que el objetivo del marco de referencia que se desarrolla como parte de esta tesis va dirigido al diseño y ejecución de cursos sobre el GSD, en donde participan estudiantes de diferentes sitios geográficos y éstos puede tener diferentes culturas y/o regirse por diferentes normas sociales, se tomará en consideración un entorno internacional para el diseño del marco de referencia propuesto. Por consiguiente, las siguientes etapas de la metodología a seguir se desarrollarán considerando el entorno internacional que pudiera afectar al diseño de la propuesta. Aunado a lo anterior, la metodología de O'Neill (2015) sugiere el diseño curricular a través de la ejecución de seis fases principales (véase Figura 3.1).



Figura 3.1. Metodología para el diseño curricular en la educación superior (O'Neill, 2015)

Estas etapas se pueden describir de la siguiente manera:

- Análisis de necesidades y filosofía educativa: El punto de partida para el diseño de algún programa, modelo, o marco de referencia educativo consiste en determinar, en primera

instancia, una necesidad real, ya sea de la sociedad o de una institución educativa, que motive su desarrollo, considerando los programas o planes de estudios existentes sobre el área que se pretende explorar. Por otro lado, también es necesario determinar un conjunto de creencias y valores a las que aspiran los estudiantes. Finalmente, una vez realizada esta actividad se determinará si existe realmente una necesidad para crear el marco de referencia y se definirán los valores y creencias del programa. Es importante mencionar que para determinar la filosofía educativa que seguirá el marco de referencia se tomarán en cuenta los principios del ABP.

- **Análisis de modelos curriculares:** Posterior a la definición de la filosofía educativa, la metodología sugiere analizar los modelos curriculares existentes que hayan sido probados con éxito en la enseñanza del tema de estudio. De esta manera, se podrán determinar aquellos aspectos (métodos y estrategias de enseñanza, aprendizaje y evaluación) que pudieran ser tomados en cuenta para su implementación en el marco de referencia que se pretende desarrollar.
- **Definición de objetivos y resultados:** Durante esta actividad se determinarán los resultados esperados de la ejecución del marco de referencia: aptitudes, habilidades y conocimientos que se pretende que los estudiantes alcancen una vez terminado el programa establecido por el marco de referencia.
- **Organización y estructura del programa:** Durante esta fase se relacionarán de manera coherente los resultados obtenidos en las fases anteriores: necesidad que origina el marco de referencia, filosofía educativa, análisis de modelos curriculares, y objetivos y resultados esperados. Para la organización y estructura de los componentes mencionados anteriormente se tendrá en cuenta el alcance (amplitud y profundidad del contenido educativo del marco de referencia), la secuencia (el orden en el que se desarrollaran las actividades propuestas en el marco de referencia), continuidad (hace referencia al contenido del programa con relación a las habilidades de los estudiantes), integración (aplicar la teoría a la práctica en diversos entornos), articulación y balance (equilibrar el conocimiento del tema de estudio con otras habilidades).
- **Diseño de estrategias de enseñanza, aprendizaje y evaluación:** Durante esta fase se definirán los métodos y estrategias de enseñanza, aprendizaje, y evaluación con base en la filosofía educativa, los objetivos y resultados esperados de la aplicación del marco de referencia propuesto. De esta manera, para el diseño del marco de referencia propuesto en esta tesis se contemplarán los principios del ABP para el diseño de las estrategias de enseñanza, aprendizaje y evaluación.
- **Diseño del módulo mezclado:** El aprendizaje mezclado incluye la forma tradicional de enseñanza en donde el profesor tiene contacto directo con los estudiantes, pero añade métodos de trabajo en línea por medio del internet y las TI. En este contexto, durante esta fase se integrarán los componentes obtenidos en las fases anteriores y se implementarán algunos conceptos del aprendizaje mezclado a través de la herramienta computacional que se desarrollará como parte de esta tesis. Una vez terminada esta fase se obtendrá el marco de referencia en sí.

Aunado a lo anterior, la metodología sugiere diseñar una estrategia de evaluación que permita monitorear y evaluar los objetivos y resultados esperados de la ejecución del marco de referencia

propuesto. Esta estrategia de evaluación se llevará a cabo durante el diseño del marco de referencia, en su implantación en un curso académico, y después de su implantación en un entorno académico real. Así mismo, durante la ejecución del curso puede ocurrir que tanto los estudiantes como los profesores necesiten de ayuda para llevar a cabo las actividades propuestas por el marco de referencia, por lo que será importante brindar apoyo académico y tecnológico.

3.1.1. Análisis de necesidades y filosofía educacional

3.1.1.1. Análisis de necesidades

La IS es una disciplina que ha evolucionado a través del tiempo, dado que inicialmente se centraba únicamente en la estructura del producto de software y de los procesos requeridos para crearlo. En este contexto, el modelo de desarrollo en cascada llegó a representar en gran medida a los conceptos elementales de la IS. Sin embargo, aunque dicho modelo representaba de manera correcta la forma en que se desarrollaban algunos productos de software, éste resultaba ser demasiado rígido. Como consecuencia de lo anterior, surgieron formas más flexibles para desarrollar el software, las cuales establecieron un proceso de varias fases que mantenía cierto grado de disciplina y estructura, tal es el caso de los modelos basados en prototipos.

No obstante, con el surgimiento del internet en la década de 1990 surgió la necesidad de crear nuevos modelos de desarrollo que fueran más flexibles e iterativos. Estos fueron caracterizados por términos como desarrollo rápido de aplicaciones (RAD, por sus siglas en inglés), que sugerían la necesidad de incluir al cliente en el proceso de desarrollo. Así mismo, surgieron nuevos modelos denominados ágiles, los cuales consideran que algunas actividades del proceso, como la documentación extensa, carecen de importancia, y los *stakeholders* y sus interacciones suelen tener una mayor prioridad.

En consecuencia, aunque hoy en día los ingenieros de software siguen realizando muchas de las mismas actividades que se realizaban en la década de 1960, lo hacen en un contexto totalmente diferente debido a que los diferentes modelos organizacionales de las empresas de desarrollo de software afectan directamente la forma en que éste se produce. Por ejemplo, se ha observado una clara tendencia de la industria de software de incorporar al enfoque de GSD como estrategia de desarrollo. No obstante, a pesar de que esto evidencia que la IS ha evolucionado, también es importante resaltar que es necesario enfocarse en mejorar los planes de estudio y la forma de enseñarla en las universidades, las cuales deben adaptarse a estas nuevas necesidades de la industria.

En relación con lo anterior, las Guías del Plan de Estudios para Programas de Pregrado en Ingeniería de Software (SE 2014), creadas por la ACM y el IEEE, mencionan que es importante tomar en cuenta al GSD y la evolución de la IS al momento de diseñar cualquier plan de estudio relacionado con programas que aborden las áreas de la misma (Ardis, Budgen, Hislop, Offutt, Sebern, y Visser, 2015). En este sentido, SE 2014 es un documento que proporciona orientación a las instituciones académicas y agencias de acreditación acerca de lo que debería constituir un programa orientado a la enseñanza de la IS a nivel licenciatura. Así, este documento establece un conjunto de recomendaciones basadas en las características que deberían poseer los egresados de algún programa relacionado con la IS y presenta dos principales aportaciones: la Guía de Conocimiento sobre la Educación de la Ingeniería de Software (SEEK, por sus siglas en inglés) que resume el conocimiento que un egresado de un programa relacionado con la IS debe poseer, y un conjunto de recomendaciones sobre las formas en que el conocimiento y las habilidades fundamentales de la IS pueden enseñarse a nivel licenciatura. La primera versión de este conjunto de recomendaciones fue publicada en 2004; sin embargo, en 2010 el equipo que trabajó en esta primera versión analizó el estado actual de la enseñanza de la IS y decidió realizar algunas actualizaciones al

documento. En este sentido, los cambios más significativos son: el reconocimiento de los modelos de ciclo de vida del software alternativos y un mayor énfasis en los requisitos y la seguridad del software.

Por otro lado, el Capítulo 3 de SE 2014, referente a las principales recomendaciones sobre la enseñanza de la IS, hace énfasis en la importancia de que los estudiantes sean preparados apropiadamente con las habilidades y conocimientos necesarios para incorporarse a la industria al momento de egresar de la universidad. Así mismo, esta guía hace referencia a la importancia de las habilidades de trabajo en equipo que los estudiantes deberían desarrollar durante su formación universitaria. Considerando lo anterior, el Capítulo 4 del mismo documento hace referencia a que es importante que los estudiantes desarrollen las habilidades para trabajar en equipo, así como para comunicarse con sus compañeros. De manera similar, el Cuerpo de Conocimientos de la Ingeniería de Software (SWEBOK) menciona la importancia de las habilidades de comunicación y trabajo en equipo para los ingenieros de software, debido a que en la práctica profesional los egresados pueden trabajar en equipos de GSD, en donde, como integrantes de un equipo, se encuentran distanciados geográficamente y pueden tener culturas diferentes; por lo tanto, el estudiante debe estar preparado para trabajar en estos grupos multiculturales. Así mismo, éste también menciona la tendencia actual hacia el GSD en la industria de software, por lo cual sugiere que los egresados posean las habilidades suaves de trabajo en equipo y comunicación que les permitan trabajar en estos entornos globales (Bourque y Fairley, 2014).

Por su parte, el Consejo Regional para el Desarrollo de la Educación y la Sustentabilidad (CONREDES) realizó un análisis acerca de la demanda de talento por parte de la industria mexicana hacia los estudiantes universitarios (Consejo Regional para el Desarrollo de la Educación y la Sustentabilidad, 2017). Dicho análisis sugiere que la industria está requiriendo actualmente a estudiantes que posean las habilidades técnicas propias de la gestión de proyectos, así como las habilidades suaves de comunicación y trabajo en equipo. Lo anterior muestra que existe claramente una necesidad en la industria actual de software para que los estudiantes estén altamente preparados, no solamente con conocimiento técnico, sino también con las habilidades suaves que les permitan trabajar en la industria. SE 2014 y SWEBOK hacen referencia concretamente a la importancia de las habilidades de trabajo en equipo y comunicación, debido a la tendencia de la industria por el GSD, la externalización de actividades y la distribución del software en diferentes partes del mundo.

Por otro lado, se realizó un análisis sobre la cobertura del GSD en universidades tanto de México como del extranjero con el objetivo de identificar el estado actual de la educación en este tópico y estudiar la importancia y necesidad de aportar nuevas propuestas o herramientas que ayuden a la enseñanza del GSD y, por otro lado, recolectar aspectos relevantes que pudieran incluirse en el marco de referencia propuesto en esta tesis. En este contexto, el análisis incluyó 61 universidades mexicanas y 60 universidades extranjeras que ofertan programas de licenciatura relacionados con la IS, tales como: Licenciatura en Ingeniería de Software, Licenciatura en Sistemas Computacionales, Ingeniería en Computación, Ingeniería en Ciencias de la Computación, y Licenciatura en Desarrollo de Software (ofrecida en instituciones extranjeras principalmente). Así mismo, en el análisis se incluyeron algunos programas de maestría relacionados con la IS ofrecidos por estas universidades.

En este contexto, el análisis evidenció que, a pesar de la importancia que hoy en día tiene el GSD en la industria de software y de las recomendaciones realizadas por parte del SE 2014 y SWEBOK sobre considerar al GSD en la educación universitaria, las universidades continúan sin incluir formalmente en sus planes de estudios a dicho tópico. Así pues, de las 61 universidades analizadas en México se identificó que el 65% no incluye tópicos relacionados con el GSD en sus

planes de estudio, perfil de egreso de sus estudiantes, misión, visión u objetivos del programa. Por otra parte, tan solo el 6% de las universidades incluyó en su plan de estudios algún tópico relacionado con el GSD, generalmente aspectos relacionados con generar en los estudiantes las habilidades para desempeñarse en entornos internacionales y multiculturales. De manera similar, el 19% menciona dentro del perfil de egreso que los estudiantes poseerán las habilidades necesarias para desempeñarse en entornos multiculturales y globales al terminar sus estudios. Finalmente, el 7% de las universidades incluyó aspectos relacionados con el GSD como parte de los objetivos del programa y el 3% mencionan aspectos relacionados con el GSD dentro de la misión o visión del programa. En este contexto, la Figura 3.2 resume los resultados obtenidos.

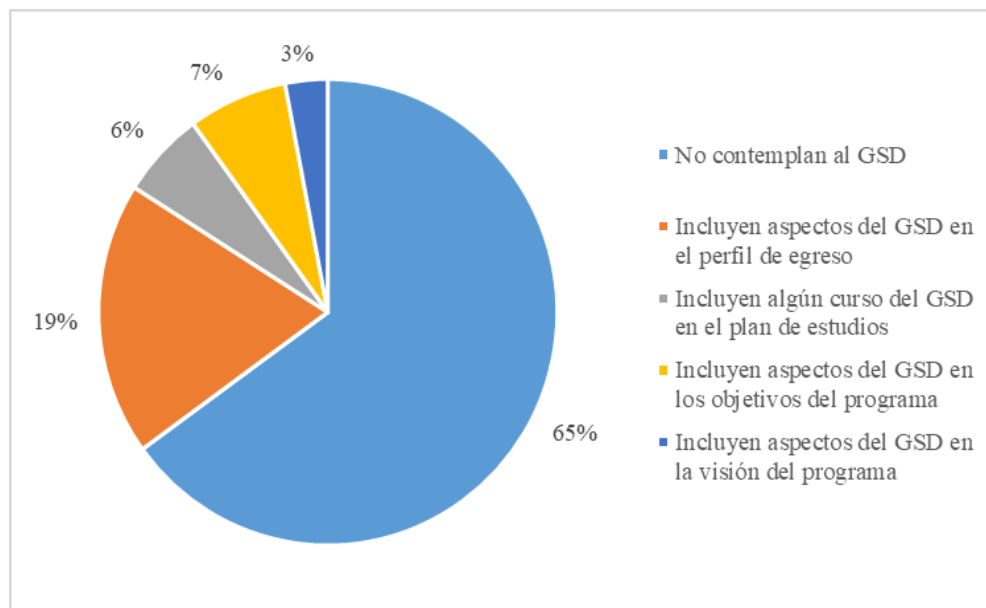


Figura 3.2. Cobertura sobre la enseñanza del GSD en las universidades de México

En el ámbito internacional, el tópico de GSD ha recibido una mayor atención por parte las universidades, puesto que el 33% de éstas incluye en los objetivos del programa la preparación de los estudiantes en temas relacionados con el GSD tanto como el desarrollo de habilidades de comunicación en ámbitos globales y multiculturales. Así mismo, es importante mencionar que el 17% de las universidades analizadas ofertan cursos de especialización sobre el GSD como parte de la preparación “adicional” de los estudiantes. Estos cursos suelen involucrar el desarrollo de proyectos con la industria en entornos globales o clases de especialización en las cuales los estudiantes aprenden a trabajar en entornos globales. Por otra parte, el 7% del total de las universidades oferta cursos relacionados con el GSD dentro de sus planes de estudio y el 43% no incluyen algún tópico relacionado con el GSD (véase Figura 3.3).

En resumen, se puede apreciar que en el ámbito nacional el GSD no es considerado parte importante de la formación de los estudiantes; no obstante, existe evidencia de la importancia de este tópico en la actual industria de desarrollo de software a nivel mundial. En contraparte, en el ámbito internacional el GSD ha tomado mayor relevancia, al grado que las universidades ofertan cursos de especialización para estudiantes de licenciatura en esta área. En este contexto, es importante resaltar que durante el análisis se pudo observar una mayor tendencia por la enseñanza del GSD a nivel maestría, puesto que diversas universidades ofertan maestrías totalmente enfocadas al GSD.

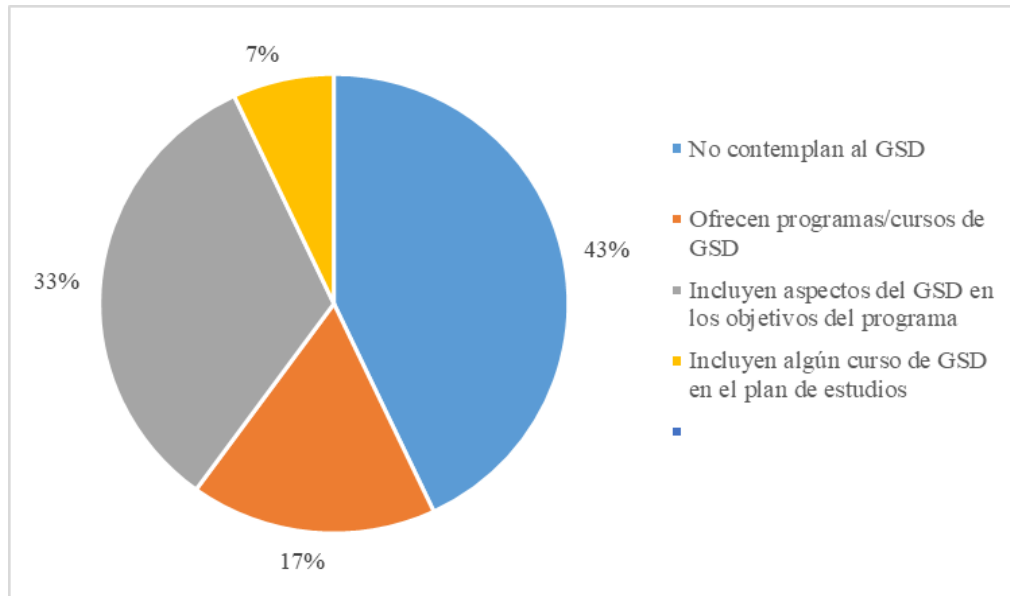


Figura 3.3. Cobertura sobre la enseñanza del GSD en las universidades internacionales

Derivado del análisis anterior se puede deducir que aunque el GSD ha tomado mayor relevancia durante los últimos años en la industria del software y que documentos oficiales como el SE 2014 y SWEBOK hacen recomendaciones acerca de la enseñanza de este tópico en las universidades, éstas no han tomado en consideración la enseñanza de este tópico dentro de sus planes de estudio, por tal motivo, es importante ofrecer alternativas como la propuesta en esta tesis para fomentar el desarrollo de habilidades suaves desde el punto de vista del GSD y enfocadas en la formación universitaria de los estudiantes.

3.1.1.2. Filosofía educacional

De acuerdo con O'Neill (2015), definir una filosofía educacional permite comunicar tanto a estudiantes como a profesores la justificación de enfoques de enseñanza, aprendizaje y evaluación presentes en el marco de referencia. En este contexto, una filosofía educacional es un enunciado que establece el propósito del marco de referencia, los valores a los que aspiran los estudiantes, la naturaleza del ambiente de aprendizaje para los estudiantes y los enfoques clave para la enseñanza, aprendizaje y evaluación. Teniendo esta idea en mente, el siguiente enunciado representa la filosofía educativa del marco de referencia propuesto en esta tesis:

“Este marco de referencia está diseñado para organizar cursos de Ingeniería de Software enfocados al GSD, orientando la preparación de los estudiantes a la adquisición y mejoramiento de las habilidades de comunicación y trabajo en equipo que les permitan desempeñarse en un entorno globalizado de desarrollo de software. Por lo tanto, se alienta a los estudiantes a ser activos y constructores de su propio conocimiento mediante la aplicación práctica de los conceptos teóricos del GSD. Así pues, el objetivo final reside en proporcionar un entorno de aprendizaje práctico que estimule a los estudiantes en la generación de experiencias que mejoren sus habilidades suaves con relación al GSD y por medio del desarrollo de proyectos prácticos de desarrollo de software. Como resultado de este enfoque de aprendizaje, el marco de referencia considera la implementación de la teoría constructivista de aprendizaje a través de la ejecución del enfoque de aprendizaje basado en proyectos. Por lo tanto, se aplicarán proyectos prácticos y se utilizarán enfoques de evaluación que

permitan medir, por un lado, el desarrollo de las habilidades de los estudiantes (comunicación y trabajo en equipo) y, por otro lado, su desempeño durante el desarrollo de los proyectos”.

3.1.2. Análisis de modelos curriculares

Una vez que se ha determinado que existe una necesidad que sustenta el desarrollo del marco de referencia propuesto y que se ha definido la filosofía educativa que regirá su diseño, se deberá realizar un análisis sobre modelos curriculares que pudieran aportar algunos aspectos educativos al mismo, puesto que, de acuerdo con O’Neill (2015), los modelos curriculares pueden servir de apoyo a los diseñadores de marcos de referencias, planes de estudio, etc., para trazar de manera sistemática y transparente los fundamentos que sustenten el uso de determinados enfoques de enseñanza, aprendizaje y evaluación. En este contexto, dos de los principales modelos curriculares utilizados para definir programas educativos son:

- El modelo de producto: Enfocado principalmente a los planes y objetivos de aprendizaje (véase Figura 3.4).
- El modelo de procesos: Enfocado principalmente a las actividades y los resultados adquiridos a partir de las experiencias del trabajo y de la vida (véase Figura 3.5).

Así, el modelo de producto es concebido como aquel que conduce a la obtención de algún producto final deseable (i.e., conocimiento de ciertos hechos, dominio de habilidades y competencias específicas, adquisición de aptitudes y valores apropiados, etc.). Así mismo, este tipo de modelos curriculares se rigen por objetivos de comportamiento. Por otro lado, el modelo de procesos hace énfasis en el desarrollo continuo y los resultados del aprendizaje se perciben en términos del desarrollo de ciertos procesos y potenciales deseables (formas de pensar, actuar y sentir) (Sheehan, 1986).

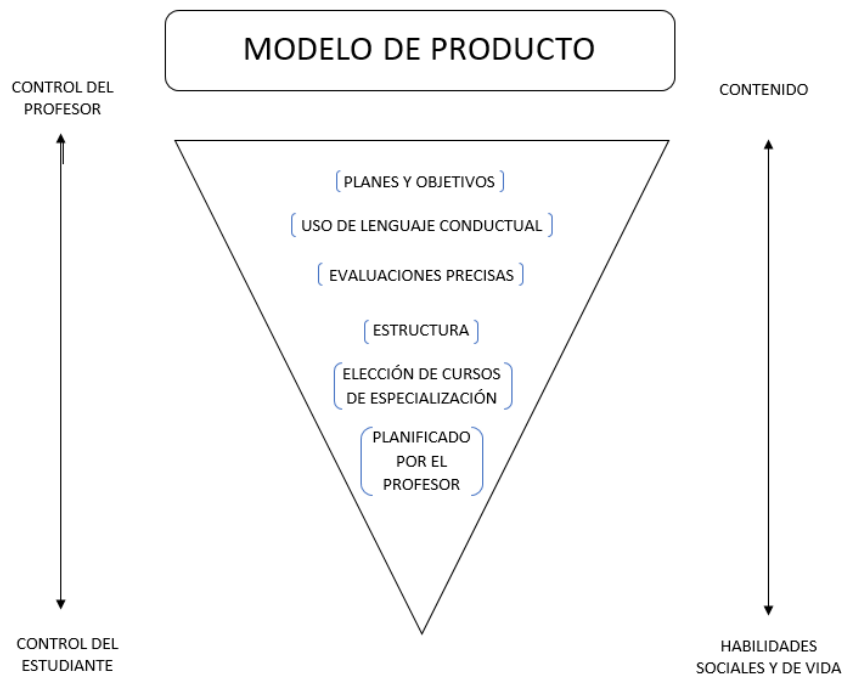


Figura 3.4. Modelo de producto para el diseño curricular (O’Neill, 2015)

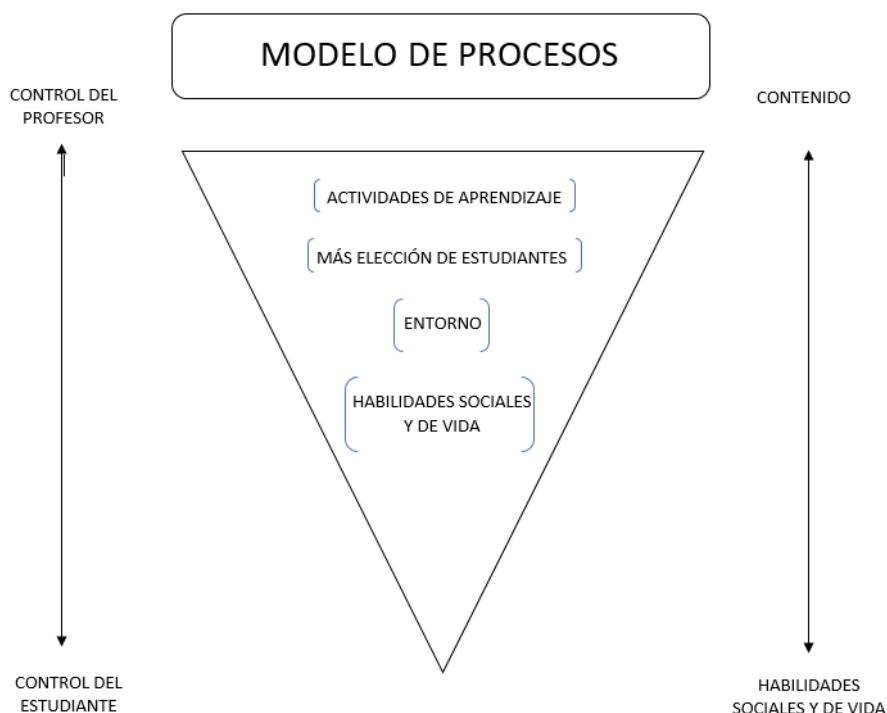


Figura 3.5. Modelo de procesos para el diseño curricular (O’Neill, 2015)

Además de los modelos citados anteriormente, existe una variedad importante de modelos curriculares específicos derivados del modelo de productos y del modelo de procesos como se aprecia en la Tabla 8.

Tabla 8. Modelos curriculares derivados del modelo de producto y del modelo de procesos (O’Neill, 2015).

Modelo de productos	Modelo de procesos
Modelo técnico/científico	Modelo no técnico
El diseño hacia atrás	Modelo de reflexión: Enfoques conversacionales
Modelo de pensamiento cognitivo	Modelos del post positivismo
Enfoque de rendimiento o sistemas. Planes de estudio tradicional o disciplinado	Relevancia personal o experiencia
Enfoque cognitivo	Enfoque crítico social
Enfoque social constructivista	Diseños centrados en el alumno
Diseños centrados en el tema	Diseños centrados en problemas
Alineamiento constructivo	Plan de estudios negociado
Modelos basados en competencias	Plan de estudios basado en el proceso
Diseño de correlación	Diseño integrado del currículo
Currículo basado en temas	Modelos basados en problemas

Con respecto a lo anterior, O’Neill (2015) argumenta que en el *modelo técnico/científico*, el desarrollo curricular es considerado un plan útil para estructurar el entorno de aprendizaje. Así

mismo, menciona que dicho modelo es descrito como un enfoque lógico, eficiente y efectivo en la entrega de educación. Por el contrario, el *modelo no técnico* es descrito como un enfoque subjetivo, personal, estético y centrado en el alumno. Así, el enfoque principal de este modelo no es el contenido o los objetivos de aprendizaje, sino el alumno. Lo anterior debido a que el modelo sigue la idea de que los componentes del programa o marco de referencia desarrollado solamente son útiles en la medida en la que el estudiante puede encontrarles utilidad para sí mismo.

En este sentido, derivado del *modelo técnico/científico* surge el *modelo de diseño hacia atrás* el cual es utilizado principalmente en el diseño de programas y planes de estudio a nivel posgrado, puesto que se vincula con la idea de atributos y competencias requeridas para estudiantes de posgrado. De manera similar, del *modelo no técnico* se derivan otro conjunto de modelos curriculares generalmente centrados en el alumno, como son el caso de los *modelos de reflexión* y los *modelos del post positivismo*. Para el caso de los *modelos de reflexión* se argumenta que éstos sugieren libertad total para que los estudiantes elijan de acuerdo con sus necesidades lo que les gustaría aprender. De este modo, el modelo sugiere que el profesor dé a conocer sus ideas de enseñanza a los estudiantes y juntos planifiquen el rumbo del curso, realizando un proceso en el cual los estudiantes proporcionen realimentación del curso al profesor y ajusten constantemente el plan educativo. Por otro lado, en el *modelo post positivismo* se sugiere menor participación de los profesores en la organización del curso. En este modelo, a los estudiantes no se les presentan ideas o contenidos para que ellos definan lo que deberían aprender, sino que se les proporcionarán contenidos y temas para que ellos busquen más información y exploren nuevas formas de aprendizaje.

Con base en lo anterior, el marco de referencia propuesto en esta tesis tomará aspectos relevantes del modelo de producto y de procesos mencionados anteriormente que se consideran importantes debido al propósito de este proyecto. En este contexto, del modelo de producto se considera importante establecer y definir de manera puntual los planes y objetivos de aprendizaje que se pretenden alcanzar. Así mismo, se considera importante seguir el principio de realizar evaluaciones precisas con relación al mejoramiento de las habilidades de los estudiantes que se pretenden mejorar con la aplicación del marco de referencia. Por otro lado, es importante que el enfoque de aprendizaje se centre en el estudiante más que en el profesor, por tal motivo, los aspectos del modelo de procesos, que dan mayor énfasis a los estudiantes, serán tomados en cuenta en el diseño y desarrollo del marco de referencia.

Aunado a lo anterior, se seguirán los principios del enfoque de enseñanza ABP derivado de la teoría constructivista de aprendizaje, la cual sustenta el principio del modelo de procesos en donde se da mayor énfasis al proceso de enseñanza centrado en el estudiante, debido a que con este enfoque el estudiante participa en el desarrollo de proyectos reales, los cuales harán que genere experiencia y, por ende, genere su propio conocimiento.

3.1.3. Definición de objetivos y resultados

Durante esta fase se deberán definir los atributos que los estudiantes desarrollarán a través del uso del marco de referencia, así como los objetivos y resultados de aprendizaje esperados. Con relación a lo anterior, O'Neill (2015) afirma que la definición de estos conceptos ayudará a diseñar el marco de referencia, las actividades de aprendizaje y el contenido del curso. Así mismo, ayudará a comunicar a estudiantes, profesores y demás interesados el propósito, enfoque y objetivos del marco de referencia.

De acuerdo con Bowden et al. (como se citó en O'Neill, 2015), los atributos que los graduados de una universidad deben desarrollar se identifican como el conjunto de cualidades, habilidades, y

conocimientos que la comunidad universitaria establece en dicha institución. Por otro lado, Adam (como se citó en O'Neill, 2015) menciona que los objetivos de un programa educativo o marco de referencia están relacionados con la enseñanza y las ideas o intenciones de los profesores; mientras que los resultados están relacionados con el aprendizaje de los estudiantes. De manera similar, Moon (como se citó en O'Neill, 2015) argumenta que los objetivos muestran el contenido general, la dirección y el propósito del programa o marco de referencia desde el punto de vista de los profesores. A su vez, los resultados de aprendizaje son expresiones que dan una visión general de lo que un alumno habrá logrado al terminar el curso; es decir, expresan los conocimientos, habilidades, y aptitudes que los estudiantes deben poseer cuando termine la ejecución del marco de referencia.

En este sentido, en adelante se definen los atributos que los estudiantes deberán desarrollar, los objetivos del marco de referencia, y los resultados de aprendizaje esperados con la implementación del marco de referencia.

3.1.3.1. Atributos de los estudiantes

Como se mencionó anteriormente, los atributos de los estudiantes son aquellas cualidades, habilidades y conocimientos que éstos deberán desarrollar a través de la implementación del marco de referencia propuesto y, dado que la propuesta de esta tesis está enfocada en mejorar la formación de los estudiantes en el GSD, se pretende que éstos desarrollen habilidades, cualidades y conocimientos que les permitan comprender y trabajar bajo este enfoque en un entorno distribuido geográficamente. En este contexto, Casey y Richardson (2008) han sugerido que los ingenieros de software que trabajen bajo un entorno de GSD posean las habilidades de: resolución de conflictos, trabajo en equipo, y destrezas comunicativas. Por otro lado, Saldaña-Ramos et al. (2014) identificaron un conjunto de habilidades que los integrantes de equipos de GSD deberían de poseer, dentro de las cuales están la gestión de la comunicación síncrona y asíncrona, iniciativa y liderazgo, capacidad de comunicación, entre otras. De manera similar, Monasor, Vizcaíno, y Piattini (2011) mencionaron la necesidad de la capacitación de los estudiantes de GSD en habilidades de comunicación, resolución de conflictos, trabajo en equipo, habilidades de gestión, trabajo en equipos multiculturales y capacidad de liderazgo. Finalmente, el SE 2014 definió la importancia de que los estudiantes desarrollen las habilidades para trabajar en equipo, así como para comunicarse con sus compañeros.

Así pues, la Tabla 9 resume los atributos que se espera desarrollar en los estudiantes considerando las recomendaciones mencionadas.

Tabla 9. Atributos por desarrollar en los estudiantes mediante el marco de referencia propuesto.

Cualidades	Habilidades	Conocimientos
<ul style="list-style-type: none"> • Responsable • Proactivo • Colaborativo • Creativo 	<ul style="list-style-type: none"> • Trabajo en equipo • Comunicación • Colaboración • Liderazgo 	<ul style="list-style-type: none"> • Conocimiento de la diversidad cultural • Gestión de proyectos de software • Sistema de control de versiones

3.1.3.2. Objetivos del marco de referencia

Los objetivos que se pretenden alcanzar son:

- Desarrollar en los estudiantes las habilidades de comunicación y trabajo en equipo con un enfoque orientado al GSD.

- Preparar a los estudiantes para trabajar en entornos globales en el área del desarrollo de software.
- Proveer a los estudiantes de experiencias prácticas acerca del GSD.

3.1.3.3. Resultados de aprendizaje

Los resultados de aprendizaje que se pretenden alcanzar en los estudiantes son:

- Conocimiento de los principios básicos del GSD y su importancia en la industria actual del software.
- Capacidad para trabajar en equipos multiculturales.
- Capacidad para trabajar con equipos y personas ubicadas en una zona horaria distinta, pudiendo gestionar eficazmente su agenda de trabajo con el objetivo de ser productivo.
- Habilidades que le permitan comunicarse de manera efectiva con su equipo de trabajo.
- Habilidades que les permitan resolver conflictos dentro de su equipo de trabajo.
- Conceptos sobre la gestión de proyectos en el contexto del GSD.
- Capacidad de gestionar y transmitir eficazmente el conocimiento sobre los proyectos de software en los que trabaje a su equipo de trabajo.

3.1.4. Organización y estructura del programa

Hasta este punto se han definido la necesidad que origina el desarrollo del marco de referencia, la filosofía educativa que regirá su diseño, los aspectos y componentes de los modelos curriculares que servirán de apoyo en su construcción, y los objetivos y resultados de aprendizaje esperados; por tal motivo, ahora es importante organizar y estructurar de manera coherente la relación entre los componentes mencionados anteriormente.

Con el objetivo de lograr coherencia en la organización y estructura de los componentes del marco de referencia, Ornstein y Hunkins (como se citó en O'Neill, 2015) sugieren enfocarse en el alcance, secuencia, integración, y articulación del marco de referencia. En este contexto, en las siguientes secciones se definirán los puntos mencionados anteriormente y que se relacionan con el diseño del marco de referencia propuesto.

3.1.4.1. Alcance

El alcance del marco de referencia hace énfasis en la amplitud y profundidad de su contenido. En este sentido, el contenido hace referencia tanto al dominio del conocimiento (contenido académico), como a las habilidades afectivas (valores y actitudes). Lo anterior es descrito comúnmente como las habilidades genéricas (i.e., conjunto de conocimientos, habilidades, actitudes y valores relacionados entre sí y que permiten el desempeño de una persona en lo personal y social) y habilidades específicas (i.e., competencias necesarias para dominar un conocimiento y aplicarlo a un área específica) requeridas en el marco de referencia.

Con respecto a lo anterior, O'Neill (2015) argumenta que no es factible intentar cubrir un amplio espectro de conocimientos y habilidades dentro del alcance de un marco de referencia, puesto que, se corre el riesgo de que los estudiantes pierdan la oportunidad de clasificar, comparar, priorizar y criticar ideas acerca de éste. Además, la sobrecarga de información en los contenidos genera un aprendizaje superficial en los estudiantes que genera que, con el transcurso de los días, la información que fue transmitida sea olvidada.

Con relación al contenido académico, el marco de referencia propuesto comprende dos ejes principales para la enseñanza del GSD: aspectos teóricos acerca del enfoque y experiencias prácticas a través del desarrollo de proyectos siguiendo los lineamientos del ABP. En este sentido, los aspectos teóricos a cubrir como parte del proceso de enseñanza de los estudiantes son los siguientes:

- Definición y características del GSD.
- Beneficios del GSD en la industria actual del software.
- Problemas y retos presentes en el GSD:
 - Distancia geográfica.
 - Distancia cultural.
 - Distancia temporal.
- Implementación de técnicas ágiles enfocadas en la gestión de proyectos de software en un entorno de GSD.
- Conceptos básicos de herramientas y estrategias de gestión del conocimiento del proyecto.
- Conceptos básicos de herramientas y estrategias de colaboración.
- Entendimiento de la diversidad cultural.

Por otro lado, el marco de referencia propuesto considera el desarrollo de pequeños proyectos de software siguiendo el enfoque del GSD. Así, los proyectos que los estudiantes desarrollarán forman parte de la estrategia de enseñanza que establecerá el marco de referencia con el objetivo de que los estudiantes desarrollen, en un entorno práctico, las habilidades de trabajo en equipo y comunicación puesto que deberán trabajar con personas de diferente cultura y forma de trabajo. Para alcanzar dicho objetivo se considera realizar dos o tres proyectos durante el curso. En este sentido, los estudiantes formarán equipos de trabajo asumiendo roles de líder de proyectos y programadores y trabajarán en un entorno de GSD afrontando los problemas y retos de dicho enfoque. Es importante mencionar que dichos problemas serán previamente explicados en clase como parte de una formación teórica. Una vez terminado un proyecto, los estudiantes deberán cambiar roles, pudiendo así experimentar diferentes responsabilidades durante el desarrollo de proyectos de GSD, principalmente con el objetivo de generar en ellos una experiencia práctica sobre la gestión de proyectos y el liderazgo.

En resumen, el marco de referencia propuesto abarca los conceptos teóricos acerca del GSD y la importancia de las habilidades de trabajo en equipo, comunicación, y colaboración como factores de éxito en el desarrollo de proyectos globales. Aunado a lo anterior, este marco de referencia se apoya en los lineamientos del ABP como estrategia de enseñanza para que los estudiantes obtengan experiencia práctica y generen su propio conocimiento a través de la implementación en entornos prácticos de los conceptos teóricos que les son transmitidos como parte de su formación académica, desarrollando así las habilidades de trabajo en equipo y comunicación, objetivos de este trabajo.

3.1.4.2. Secuencia

La secuencia del contenido presente en el marco de referencia se ocupa del orden en el que las cosas ocurren dentro de éste. Así, la literatura hace énfasis en dos enfoques principales para organizar la secuencia del contenido: el enfoque centrado en la lógica del tema de estudio, y el enfoque centrado en la forma en la que los individuos procesan el conocimiento. En este sentido,

O'Neill (2015) menciona cuatro formas diferentes de abordar la secuencia del contenido en un marco de referencia:

- Aprendizaje simple a complejo.
- Requisitos de aprendizaje (aspectos particulares captados antes que otros).
- Todo por separado (investigación, problemas, conceptos).
- Aprendizaje cronológico (histórico, en desarrollo).

Considerando lo anterior, O'Neill (2015) menciona que los enfoques indicados anteriormente no son excluyentes entre sí. En este sentido, la propuesta de esta tesis pretende seguir un enfoque de aprendizaje simple a complejo. Así pues, al inicio del curso se introducirá a los estudiantes en los conceptos básicos del GSD para que éstos comprendan la importancia del tema en la industria actual del software. Posteriormente se enseñará a los estudiantes conceptos más específicos relacionados con la gestión de proyectos de software en entornos de GSD, así como los principios básicos acerca de las herramientas y estrategias de comunicación, colaboración y gestión del conocimiento del proyecto.

Una vez que los estudiantes conozcan los conceptos básicos del GSD, se les proporcionarán algunos proyectos de software que deberán desarrollar siguiendo el enfoque de GSD, de esta manera se implementarán los conceptos del ABP que sugieren introducir a los estudiantes en un proyecto planteándoles el enunciado del problema y que posteriormente sean ellos, junto con el profesor, quienes debatan acerca de la solución y trabajen en su desarrollo. De este modo los estudiantes aplican los conceptos teóricos en los proyectos y debido a que trabajarán en un enfoque de GSD se pretende que desarrollen las habilidades de trabajo en equipo y comunicación a través de su experiencia en los proyectos. Cabe señalar que los estudiantes desarrollarán los proyectos que el profesor les plantee y seguirán tomando clases acerca del GSD, pudiendo así resolver sus dudas en el aula con ayuda del profesor.

3.1.4.3. Integración

La integración está relacionada con la vinculación que se da entre la diferente información que forma parte del contenido, es decir, la relación existente entre cada uno de los componentes del marco de referencia para que los estudiantes obtengan una visión general de éste. Así mismo, diversos autores relacionan la integración con los vínculos entre la información y los temas que definen a un curso académico con aspectos del mundo real. En este sentido, la integración del contenido del marco de referencia está estrechamente relacionada con su importancia en el mundo real. Al respecto O'Neill (2015) sugiere que los estudiantes aprenden mejor cuando se les pide que apliquen el conocimiento y las habilidades aprendidas en diferentes lugares, asimismo, menciona que los principales enfoques que facilitan la integración son: casos de estudio, proyectos, mapas conceptuales, y evaluaciones finales (e.g., seminarios, disertaciones). Así pues, el marco de referencia propuesto está altamente relacionado con temas actuales del mundo real como es el caso del GSD y pretende relacionar a los estudiantes con el tema a través del desarrollo de proyectos reales o simulados en los que experimenten esta forma de trabajo.

3.1.4.4. Articulación

Existe la necesidad de que los estudiantes reciban algún indicador de su progreso a través del marco de referencia. Por tal motivo, la Tabla 10 muestra la relación que existirá entre los temas que comprenderá el marco de referencia y el conocimiento que obtendrá el estudiante.

Tabla 10. Relación entre temas del marco de referencia y el conocimiento del estudiante.

Tema	Conocimiento teórico	Habilidades de comunicación	Habilidades de trabajo en equipo	Gestión de proyectos en el GSD
Definición y características del GSD	✓			
Beneficios del GSD en la industria actual del software	✓			
Problemas y retos presentes en el GSD	✓			
Conceptos básicos de la gestión de proyectos de software en un entorno de GSD	✓			✓
Entendimiento de la diversidad cultural	✓	✓	✓	
Conceptos básicos de herramientas y estrategias de gestión del conocimiento del proyecto	✓	✓	✓	
Conceptos básicos de herramientas y estrategias de colaboración	✓	✓	✓	
Desarrollo de proyectos		✓	✓	✓

3.1.5. Diseño de estrategias de enseñanza, aprendizaje y evaluación

Las estrategias de enseñanza y aprendizaje que se implementarán en el marco de referencia están relacionadas con la filosofía educativa establecida anteriormente y los objetivos y resultados esperados de la ejecución de esta propuesta. Con base en lo anterior, el diseño de la estrategia de enseñanza y aprendizaje del marco de referencia considerará el ABP con el propósito de que los estudiantes comprendan la importancia del GSD y desarrollen las habilidades suaves que les permitan trabajar bajo este enfoque.

En este sentido, el ABP surge como resultado de la investigación del filósofo y educador John Dewey (1959), quien argumentaba que los estudiantes podrían obtener un aprendizaje más profundo si se involucraban en actividades, tareas y problemas que simulan lo que los expertos y profesionales realizan en situaciones del mundo real. Así mismo, en la década de los 90 diversos investigadores en el área de la educación observaron que los estudiantes perdían con facilidad la atención y motivación por aprender, puesto que se aburrían en los salones de clases. Aunado a lo anterior, la falta de motivación y el abuso en la transmisión de conocimiento genera que los estudiantes obtengan un conocimiento superficial que terminan perdiendo con el paso del tiempo. Derivado de lo anterior, algunas teorías de enseñanza-aprendizaje basadas en el constructivismo sugieren mantener la motivación de los estudiantes a través de la aplicación de entornos prácticos en donde resuelvan problemas reales, permitiendo así que estos mismos estudiantes sean generadores

de su propio conocimiento y que el conocimiento generado sea más profundo y no se pierda con facilidad con el paso del tiempo.

De esta manera, de acuerdo con Krajcik y Blumenfeld (2006), el ABP es un enfoque general que se suele utilizar en el diseño de entornos de aprendizaje y que está representado por cinco características principales:

1. Inicia con una pregunta que de manera general expresa un problema a resolver.
2. Los estudiantes exploran la pregunta inicial participando en dinámicas grupales que pretenden exponer ideas para la resolución del problema inicial. De esta manera, el estudiante aprende y aplica nuevas ideas relacionadas con la disciplina de estudio.
3. Los estudiantes, profesores y demás miembros participantes en el proceso de aprendizaje participan en actividades colaborativas con el objetivo de hallar una solución al problema planteado al inicio del curso.
4. Los estudiantes cuentan con herramientas de aprendizaje que los ayudan a participar en actividades que normalmente están más allá de sus posibilidades.
5. Los estudiantes desarrollan un conjunto de productos entregables relacionados con la solución del problema planteado inicialmente.

Por otro lado, Krajcik y Blumenfeld (2006) también mencionan que el ABP se rige bajo cuatro estrategias principales de enseñanza-aprendizaje: construcción activa, aprendizaje situado, interacciones sociales, y herramientas cognitivas. Las siguientes secciones exponen estas estrategias y su relación con el marco de referencia propuesto.

3.1.5.1. Construcción activa

Diversas investigaciones en el área de la educación sugieren que el aprendizaje profundo ocurre cuando el estudiante construye activamente su conocimiento por medio de sus experiencias e interacciones con el mundo real, mientras que el conocimiento superficial ocurre cuando el estudiante solamente obtiene información transmitida por un profesor. De esta manera, el desarrollo del conocimiento es un proceso continuo que requiere que el estudiante construya y reconstruya su conocimiento a través del conocimiento previo y sus experiencias (Kolbe y Jorgenson, 2018).

En este sentido, la filosofía educativa establecida anteriormente sugiere que los estudiantes sean activos y constructores de su propio conocimiento, además, dentro de los objetivos del marco de referencia se estableció que se proveerá al estudiante de experiencias prácticas en el contexto del GSD. Por tal motivo, la estrategia de enseñanza a través de la construcción activa del conocimiento se ajusta con las necesidades del marco de referencia propuesto y será utilizada como parte de las estrategias de enseñanza implementadas en éste.

3.1.5.2. Aprendizaje situado

En el ABP se sugiere que el aprendizaje efectivo ocurre de mejor manera cuando se sitúa al estudiante en un entorno auténtico⁵ del mundo real. Esta estrategia permite al estudiante identificar

⁵ De acuerdo con Stein (1998), el aprendizaje situado permite a los estudiantes crear conocimiento a partir de las actividades reales de la vida diaria. En este sentido, el aprendizaje situado consiste en incorporar el tema de estudio en las experiencias cotidianas de los alumnos y en crear oportunidades para que los alumnos experimenten con dicho tema

la importancia del tema de estudio y las actividades y tareas que lleva a cabo. Además, el estudiante es capaz de identificar la importancia del tema para él y su comunidad, así mismo aprende como debe aplicar sus actividades para resolver problemas del mundo real de acuerdo con el contexto del tema de estudio. De manera similar, esta estrategia permite que los estudiantes relacionen el conocimiento previo con el nuevo conocimiento y su aplicación en diferentes contextos (Buchanan, 2018).

En relación con lo anterior, uno de los objetivos del marco de referencia es preparar a los estudiantes para trabajar en entornos globales en el área del desarrollo de software y se pretende que éste sea alcanzado a través de esta estrategia de aprendizaje en donde el estudiante experimente las actividades del GSD en un entorno real o simulado. Por lo tanto, se pretende que con la aplicación de esta estrategia se alcancen los siguientes resultados de aprendizaje:

- El estudiante conocerá, a través del aprendizaje situado, los principios básicos del GSD y su importancia en la industria actual del software.
- El estudiante será capaz de trabajar en equipos multiculturales, debido a que podrá experimentar de manera cercana el trabajo con personas de diferentes culturales, formas de trabajo y, posiblemente, hablas.
- El estudiante será capaz de trabajar con equipos y personas ubicadas en una zona horaria distinta a la suya, pudiendo gestionar eficazmente su agenda de trabajo con el objetivo de ser productivo.
- El estudiante aprenderá los conceptos de la gestión de proyectos en el contexto del GSD, puesto que trabajará en un entorno de GSD y aplicará estos conceptos.

De manera similar, la filosofía educativa establece que se proporcionará al estudiante un entorno de aprendizaje práctico en donde experimente el desarrollo de proyectos bajo el enfoque del GSD, y precisamente a través del aprendizaje situado es que se plantea el alcanzar esta meta. Por tal motivo, la estrategia de aprendizaje situado será implementada en el marco de referencia propuesto.

3.1.5.3. Interacciones sociales

El enfoque del ABP hace énfasis en la importancia de las interacciones sociales en el proceso de aprendizaje, debido a que se considera que se obtienen buenos resultados de la interacción social entre el profesor, estudiantes y miembros de la comunidad que trabajan en conjunto para realizar ciertas actividades que fomenten la construcción del conocimiento compartido. Así, los estudiantes desarrollan conocimiento a través de compartir, usar, y debatir ideas con sus compañeros (Krajcik y Blumenfeld, 2006).

En este contexto, esta estrategia de enseñanza está correctamente alineada con la filosofía educativa establecida anteriormente, debido a que ésta sugiere preparar a los estudiantes en habilidades de trabajo en equipo y comunicación, lo cual se puede lograr a través de la interacción entre los estudiantes puesto que deberán trabajar en equipo y establecer mecanismos de comunicación correctos para desarrollar los proyectos que se les indique. De manera similar, uno de los objetivos del marco de referencia es precisamente desarrollar en los estudiantes dichas habilidades enfocadas al GSD, por esta razón, esta estrategia puede ser correctamente aplicada. Por

en el contexto de los desafíos del mundo real; así, se adquiere conocimiento al mismo tiempo que se realiza el aprendizaje del aula al ámbito de la práctica.

otro lado, dentro de los resultados de aprendizaje esperados que están relacionados con esta estrategia se encuentran los siguientes:

- El estudiante será capaz de trabajar en equipos multiculturales.
- El estudiante desarrollará las habilidades que le permitirán comunicarse de manera efectiva con su equipo de trabajo.

Lo anterior evidencia la necesidad de aplicar esta estrategia en el contexto del marco de referencia propuesto, con el objetivo de alcanzar los objetivos y resultados esperados del mismo.

3.1.5.4. Herramientas cognitivas

Dentro del ABP se resalta la importancia de las herramientas en el proceso de aprendizaje. El uso de estas herramientas pudiera ayudar a expandir lo que los estudiantes pueden aprender. En este sentido, diversas herramientas computacionales (i.e., software) pueden considerarse como “herramientas cognitivas”, debido a que éstas permiten que los estudiantes desarrollen diversas tareas complejas que resultarían difícil de realizar de manera manual. Además, estas herramientas pudieran servir de apoyo también a los profesores para organizar el contenido del curso, asignar tareas, entre otras cosas.

Con relación a lo anterior, esta tesis propone el desarrollo de una herramienta computacional que sirva de apoyo a estudiantes y profesores para organizar y llevar a cabo cursos sobre el GSD, siguiendo los lineamientos establecidos por el marco de referencia propuesto. En este sentido, uno de los objetivos establecidos en el marco de referencia es que los estudiantes desarrollen las habilidades de comunicación y, por otro lado, se pretende también que los estudiantes desarrollen colaborativamente los proyectos que se les asignen durante su formación. Así, dentro de la herramienta de apoyo que se desarrollará se implementará las CWS, con el objetivo de que los estudiantes aprendan a identificar momentos oportunos de colaboración con sus compañeros, a través del monitoreo constante del proyecto. De esta manera, el estudiante será capaz de detectar, identificar, o crear oportunidades para la interacción, comunicación, y colaboración con sus compañeros. Aunado a lo anterior, el estudiante que funja como líder de proyectos podrá monitorear el entorno del desarrollo con el objetivo de supervisar y controlar el desarrollo del proyecto y de su equipo, dado que se añadirá una capa extra al modelo de CWS. En este contexto, la Figura 3.6 muestra el modelo de CWS que se implementará en la plataforma, y más adelante se detallará su implementación formal.

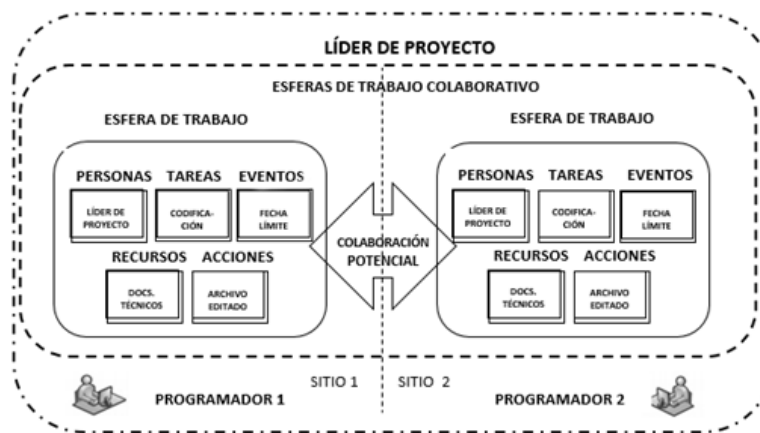


Figura 3.6. Modelo de CWS que será implementado en la herramienta cognitiva (plataforma computacional)

3.1.5.5. Estrategias de evaluación

El objetivo de las evaluaciones es monitorear el aprendizaje y desarrollo de habilidades suaves en los estudiantes. En este sentido, se ha mencionado anteriormente que los estudiantes desarrollarán proyectos de GSD como parte de su formación y se han sugerido evaluaciones constantes siguiendo los principios de las metodologías ágiles de desarrollo. Por lo tanto, se sugiere que para la entrega de los avances (*sprints* en *Scrum*) del proyecto, el profesor evalúe el desempeño de los estudiantes en relación con el trabajo en equipo, su nivel de comunicación, el trabajo del jefe proyectos, y como tal el resultado del trabajo de los estudiantes. Considerando este objetivo, el profesor hará uso de la herramienta computacional que también es propuesta en esta tesis, a través de la cual podrá revisar archivos de registros de información del trabajo de los estudiantes, además que podrá monitorear en cualquier momento su desempeño.

Aunado a lo anterior, el profesor podrá realizar exámenes de conocimientos en el aula de clases si lo considera necesario.

3.2. Diseño y construcción de la plataforma tecnológica

Una vez que se ha definido el marco de referencia que establece las pautas y lineamientos para el desarrollo de cursos enfocados al GSD, es importante contar con una herramienta que automatice y facilite su ejecución y seguimiento. Por tal motivo, esta tesis plantea, además del marco de referencia, el desarrollo de una plataforma computacional basada en *web* que sirva de apoyo a universidades, profesores y estudiantes en el desarrollo de cursos de GSD tomando como base lo establecido en dicho marco de referencia.

La plataforma que lleva por nombre Cadxela GSD (o Descubre GSD, en su traducción de Zapoteco a Español) se ha creado como una de las herramientas cognitivas que sugiere el ABP para apoyar al estudiante en la generación de conocimiento. En este sentido, el marco de referencia propone la ejecución de pequeños proyectos de software que se rijan bajo el enfoque del GSD y que permitan al estudiante, a través del concepto del aprendizaje situado, generar conocimiento teórico-práctico acerca de este enfoque de desarrollo. De esta manera, Cadxela GSD sirve de apoyo en la configuración y puesta en marcha de cursos de GSD, y es una herramienta que permite a los estudiantes planificar y gestionar el desarrollo de los proyectos que les son asignados.

Con relación a lo anterior, la herramienta permite alcanzar los siguientes objetivos del marco de referencia:

- Desarrollar en los estudiantes las habilidades de comunicación y trabajo en equipo con un enfoque orientado al GSD, puesto que a través de ésta se establece la comunicación tanto formal como informal. Además, se implementan conceptos de las metodologías ágiles que fomentan el trabajo en equipo y la colaboración, tales como las reuniones diarias y la planificación ágil donde participa todo el equipo de desarrollo.
- Proveer a los estudiantes de experiencias prácticas acerca del GSD, debido a que deberán planificar y gestionar el desarrollo de los proyectos que les son asignados y deberán trabajar activamente con compañeros que probablemente se encuentran en diferente zona geográfica.

Así pues, Cadxela GSD permite a las universidades, profesores, y estudiantes llevar a cabo las actividades mostradas en la Figura 3.7 y que se resumen como:

- Registro y configuración del curso.

- Registro y configuración de los proyectos.
- Seguimiento y control de los proyectos.
- Evaluación de los proyectos.
- Entrega de los proyectos.

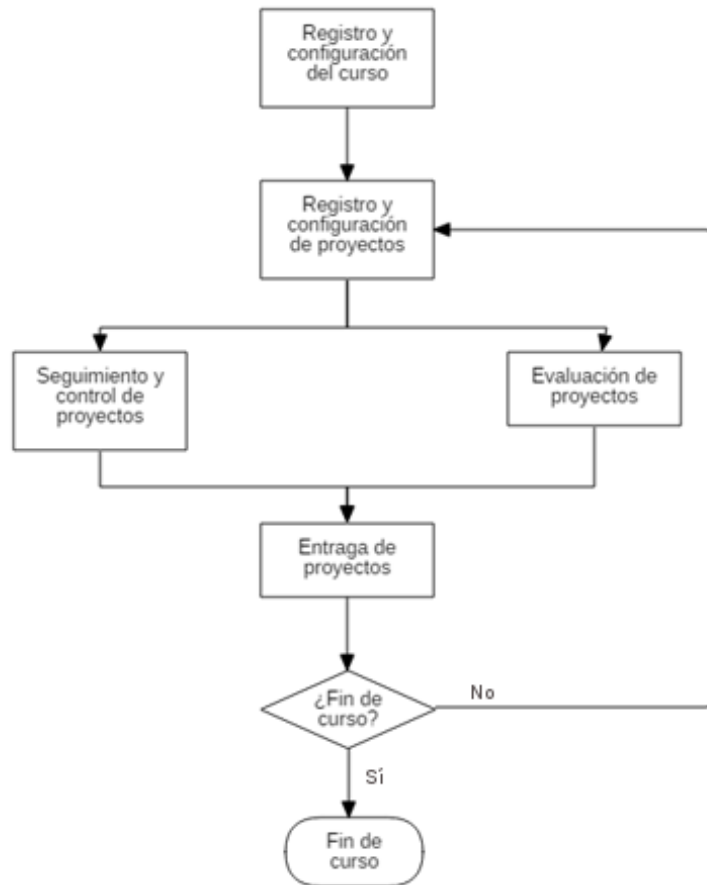


Figura 3.7. Diagrama de flujo de ejecución de un curso de GSD a través de Cadxela GSD

3.2.1. Registro y configuración del curso

Cuando dos universidades hayan decidido iniciar un curso de GSD siguiendo los lineamientos del marco de referencia propuesto en esta tesis, deberán registrar y configurar el curso. Para ello, la herramienta permite registrar la información de las universidades participantes (nombre, país, región, y zona horaria) y del curso (nombre, motivo, fechas de inicio y terminación), necesaria para la configuración de éste.

De esta manera, el sistema permite la creación de cursos de GSD siguiendo el proceso que se muestra en la Figura 3.8 y que se resume a continuación:

1. El administrador del sistema registra la información relacionada con las universidades participantes. Posteriormente, registra la información del curso a realizar, la cual es proporcionada por las universidades participantes. Por último, registra la información de los

profesores responsables del curso, la cual también es proporcionada por las universidades participantes.

- Finalmente, el sistema notifica vía correo electrónico a los profesores acerca de su registro al sistema y proporciona la contraseña de acceso para que ellos continúen con la siguiente etapa del curso, el registro, y la configuración de los proyectos.

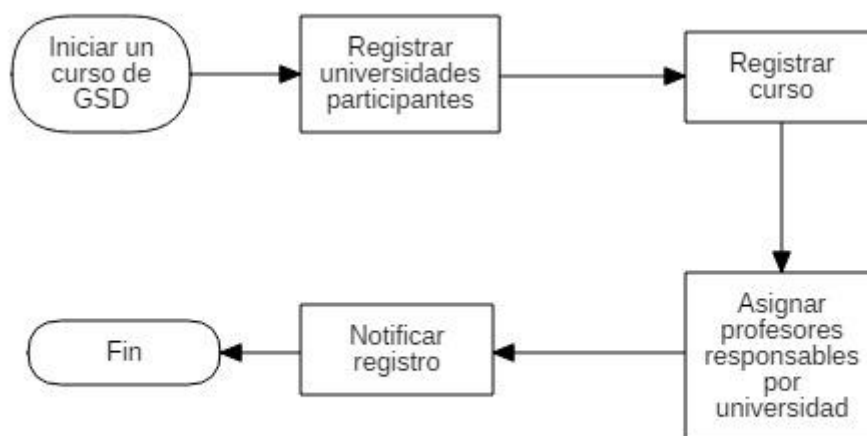


Figura 3.8. Diagrama de flujo para el registro y configuración de un curso de GSD

3.2.1.1. Desarrollo

Una vez que se ha definido el proceso a seguir para el registro y configuración de cursos de GSD, es necesario desarrollar los módulos que permitan realizar estas acciones en la herramienta. En este sentido, el desarrollo consistió en el diseño de la Base de Datos (BD) que permitió almacenar la información relacionada con las universidades, los cursos, y los usuarios. Así mismo, se codificaron tanto los módulos que permiten la conexión entre la interfaz y la BD, así como el diseño y codificación propios de la interfaz. Finalmente, la Figura 3.9 muestra las diferentes interfaces generadas para realizar las actividades para el registro y configuración del curso.

3.2.2. Registro y configuración de los proyectos

Posterior al registro y configuración del curso, es necesario que los profesores titulares de las universidades establezcan los grupos de estudiantes que participarán en éste. En este contexto, se ha restringido que por cada universidad deba registrarse solamente un grupo de estudiantes. Así pues, la herramienta permite al profesor registrar la información del grupo que participa por su institución, para después registrar la información de cada uno de los estudiantes (nombre, teléfono, y correo electrónico) que lo integran.

Siguiendo con las recomendaciones establecidas en el marco de referencia, el cual ha señalado que los estudiantes que participen en el curso deben aprender conceptos teóricos sobre el GSD, se asume que una vez que los estudiantes participantes se hayan registrado en el curso, éste se dará formalmente por iniciado. En este sentido, al inicio del curso los estudiantes deberán recibir clases teóricas sobre el GSD con el objetivo de que comprendan los conceptos básicos que les permitan desarrollar los proyectos que les sean asignados en la plataforma. Con relación a lo anterior, los

profesores deberán asignar los proyectos de GSD a los estudiantes cuando lo consideren necesario, es decir, cuando hayan cubierto los conceptos básicos del GSD.

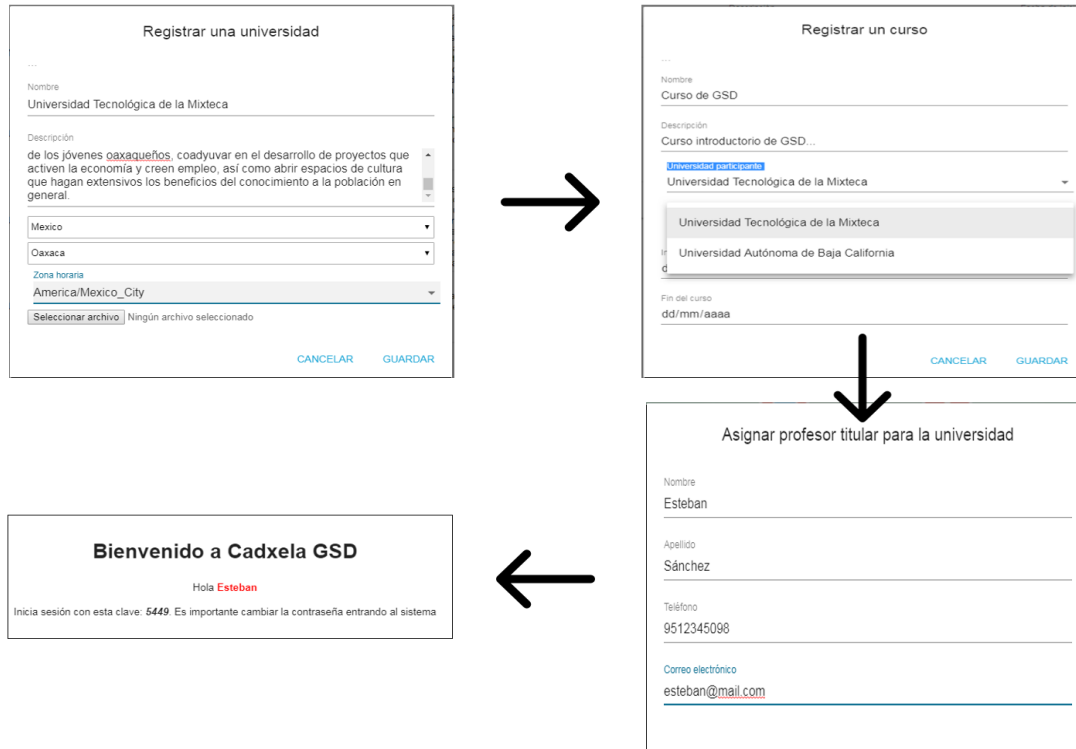


Figura 3.9. Interfaces para el registro y configuración de cursos

Con respecto a la asignación de proyectos, los profesores se apoyarán en Cadxela GSD para organizar los equipos de estudiantes y asignar los proyectos a cada uno de éstos. Cabe aclarar que es decisión del profesor la cantidad de equipos que participarán, así como la cantidad de integrantes de los mismos, siendo dos estudiantes la cantidad mínima para la formación de un equipo. Después de que se hayan formado los equipos, la herramienta permite asignar un proyecto a cada uno de éstos. Para la asignación de un proyecto es necesario contar con un nombre, descripción del proyecto, y fechas de inicio y terminación de éste. Así mismo, es necesario que el profesor asigne los roles que los estudiantes llevarán a cabo durante la ejecución del proyecto: líder de proyecto, programador, analista, y diseñador; no obstante, el rol primario para el desarrollo de proyectos es el de líder, puesto que el curso se enfoca en mejorar las habilidades suaves con relación a este papel.

Hasta este punto los equipos tienen ya información del proyecto que desarrollarán y quiénes serán sus compañeros de trabajo, pero aún hace falta establecer los requisitos del sistema. En este sentido, fuera de la herramienta, los estudiantes deben implementar conceptos de metodologías ágiles para el desarrollo de los proyectos, concretamente algunas técnicas y métodos de *Scrum*. Por lo tanto, los requisitos del sistema deben agregarse en Cadxela GSD como historias de usuario, siguiendo las recomendaciones de *Scrum*. Debido a lo anterior, es necesario que el profesor asigne las historias de usuario que los estudiantes deberán llevar a cabo para terminar el proyecto.

La elección de aplicar un enfoque ágil como *Scrum* en la realización del proyecto de GSD se fundamenta en los argumentos de Holmström, Fitzgerald, Ågerfalk, y Conchúir (2006), quienes afirmaron que un enfoque ágil ayuda a reducir los problemas relacionados con las distancias temporal, geográfica, y socio cultural, presentes en los entornos de GSD y, como consecuencia, mejoran la comunicación y el trabajo en equipo. Además, Cohn (2010) argumentó que la aplicación

de *Scrum* en los proyectos de GSD, puede ayudar a los equipos distribuidos geográficamente a desempeñarse, en términos de rendimiento, en un nivel cercano a como lo harían los equipos ubicados en un mismo sitio.

3.2.2.1. Desarrollo

Como se mencionó anteriormente, el profesor deberá registrar en Cadxela GSD a los grupos de estudiantes que participarán en el curso. Para ello, se requirió añadir a la BD la información de los grupos (i.e., nombre, universidad, miembros del grupo), equipos (i.e., nombre, curso al que pertenece, miembros del equipo, proyecto a desarrollar), estudiantes (i.e., nombre, correo electrónico, teléfono, grupo, equipo al que pertenece, y rol dentro del equipo), y proyectos (i.e., nombre, descripción, equipo que lo desarrollará, curso, fechas de inicio y terminación, historias de usuario). Posteriormente, se codificaron los módulos que permiten la conexión de la interfaz del usuario con la BD.

3.2.2.1.1. Registro de grupos de estudiantes

Para registrar a un grupo, el profesor debe establecer un nombre para éste y posteriormente agregar a los estudiantes que formarán parte del mismo, la Figura 3.10 muestra el proceso a seguir.

El diagrama muestra un flujo de dos pantallas de interfaz de usuario. La primera pantalla, titulada "Registrar un grupo", tiene un campo de texto "Nombre" con el valor "101|". Debajo del campo hay dos botones: "CANCELAR" y "GUARDAR". Una flecha grande apunta a la segunda pantalla, titulada "Agregar estudiante a este grupo". Esta pantalla tiene cuatro campos de texto: "Nombre" con el valor "Melisa", "Apellido" con el valor "Aguilar Rojas", "Teléfono" con el valor "7652341891" y "Correo electrónico" con el valor "melisa@mail.com". Debajo de los campos hay dos botones: "CANCELAR" y "GUARDAR".

Figura 3.10. Registro de grupo de estudiantes

3.2.2.1.2. Registro de los equipos de trabajo

Para realizar el registro de los equipos es necesario que los profesores de las universidades participantes hayan registrado y establecido la información de los grupos de estudiantes que participarán en el curso. En este sentido, para agregar a un equipo, cualquiera de los profesores titulares deberá presionar la opción de agregar equipo y, de la lista de estudiantes de las universidades participantes, deberá seleccionar a aquellos que serán incluidos, así como asignar un nombre al equipo. Por otro lado, solamente un profesor puede agregar o modificar la información de la configuración de los equipos y proyectos a la vez, esto con el objetivo de evitar conflictos en el registro de los datos, la Figura 3.11 muestra cómo se agrega un equipo a través de Cadxela GSD.

Figura 3.11. Registro de equipos de trabajo

3.2.2.1.3. Asignación de los proyectos y roles

Después que se hayan creado los equipos, es necesario que a éstos se les asignen los proyectos en los cuales trabajarán. Al mismo tiempo, los profesores deben llevar a cabo la enseñanza de los conceptos del GSD y, junto con los estudiantes, la gestión de los proyectos. En lo que se refiere a la asignación de los proyectos, Cadxela GSD proporciona al profesor la opción de asignar un proyecto. Para ello, los profesores deberán primeramente asignar roles a cada uno de los estudiantes del equipo, para que posteriormente registren la información del proyecto (i.e., nombre, descripción, y fechas de inicio y terminación). La Figura 3.12 muestra cómo se asignan roles y proyectos a los estudiantes.

Figura 3.12. Asignación de roles y proyectos

3.2.2.1.4. Registro de las historias de usuario

Finalmente, el profesor debe agregar las historias de usuario⁶ que resuman las funcionalidades requeridas para el proyecto. La Figura 3.13 muestra cómo se agregan estas historias de usuario.

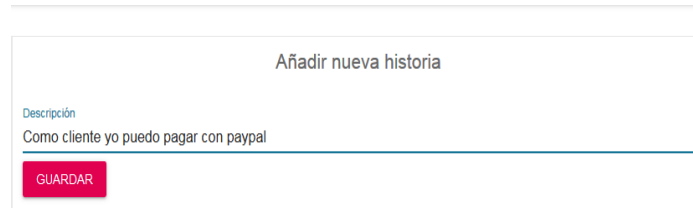


Figura 3.13. Registro de historias de usuario del proyecto

3.2.3. Seguimiento y control de los proyectos

Una vez que se ha asignado un proyecto a un equipo, es tarea de los estudiantes llevar a cabo la planificación, el desarrollo, y la gestión de éste. En este contexto, Cadxela GSD permite que el estudiante asuma el rol de líder del proyecto para realizar dos tipos de planificación con base en los lineamientos de *Scrum*: una planificación de liberación del proyecto, y la planificación de las iteraciones.

La planificación de la liberación es el proceso de creación de un plan de alto nivel que cubre un largo periodo de tiempo, durante el cual se desarrollará un proyecto de software (Cohn, 2005). Este plan transmite expectativas sobre lo que es probable que se desarrolle y en qué lapso. Aunque un plan de liberación es una excelente vista de alto nivel de cómo un equipo de desarrollo pretende entregar el producto, no permite al equipo expresar a detalle cada una de las actividades y tareas que se deben llevar a cabo para producir los entregables en cada iteración. Por tal motivo, se debe llevar a cabo un plan para cada iteración, en el cual el equipo analice de manera más detallada y enfocada lo que será necesario hacer para implementar por completo solamente las historias de usuario seleccionadas para dicha iteración.

Con respecto a lo anterior, Cadxela GSD permite al estudiante que asume el rol de líder del proyecto llevar a cabo la planificación de liberación y la planificación de las iteraciones siguiendo los lineamientos de *Scrum* como metodología ágil de desarrollo. Así, el estudiante tiene acceso a un menú de opciones para realizar acciones tales como:

- Planificación del proyecto: El estudiante realiza la planificación de liberación y de iteraciones.
- Seguimiento y control del proyecto: El estudiante tiene acceso a información relacionada con el avance real del proyecto y lo podrá comparar contra las planificaciones realizadas. Además, puede dar seguimiento a las diferentes interacciones y colaboraciones realizadas entre su equipo de trabajo.
- Iteraciones o *sprints*: El estudiante gestiona las diferentes actividades y tareas a realizar en cada iteración, además debe gestionar el estado de cada una de éstas.

⁶ De acuerdo con Cohn (2004), las historias de usuario son parte de cualquier método ágil que buscan ayudar a que el desarrollador cambie el enfoque tradicional de escribir requisitos por otro orientado a narrarlos. Una historia de usuario incluye una o dos oraciones escritas que narran la funcionalidad deseada.

- **Comunicación informal:** Con el objetivo de ayudar a los estudiantes a identificar el momento oportuno para la comunicación y colaboración informal, se ha implementado el concepto de CWS. En este sentido, la CWS permite a los estudiantes monitorear el trabajo de sus compañeros, así como si están disponibles o no para realizar una interacción. De esta manera, Cadxela GSD permite a los estudiantes monitorear el estado de sus compañeros, para así determinar si es posible realizar una solicitud de comunicación.
- **Estadísticas del proyecto:** El estudiante tiene acceso a datos relevantes del proyecto mediante gráficas de iteraciones, reuniones programadas, e información sobre las historias de usuario.
- **Calendario de actividades:** El estudiante tiene acceso a un calendario de trabajo en donde se refleja información relevante del proyecto, como el inicio y fin del curso, el inicio y fin del proyecto, y sobre las iteraciones planificadas, además, tiene acceso a notas publicadas por el profesor sobre asuntos importantes o eventos que se llevarán a cabo.
- **Reuniones:** Por último, el estudiante tiene la opción de planificar una reunión formal con los integrantes de su equipo y obtener un reporte de las reuniones que se han planificado.

Los estudiantes con un rol diferente al de líder del proyecto deberán participar activamente en la planificación del proyecto durante las diferentes reuniones, tanto formales como informales que se lleven a cabo. Sin embargo, el menú de opciones para esta clase de usuarios se limita a ver la información relacionada con las iteraciones o *sprints*, a establecer una comunicación informal, a consultar las estadísticas del proyecto, a revisar el calendario de actividades, y a participar en las reuniones. De esta manera, solamente el estudiante con el rol de líder del proyecto tendrá acceso a la planificación del proyecto y al seguimiento y control del mismo. No obstante, el marco de referencia sugiere que se realicen diferentes proyectos durante la ejecución del curso con el objetivo de formar diferentes equipos de estudiantes. Esto permite que se realicen múltiples interacciones que podrían generar diferentes problemas de comunicación con los que los estudiantes deberán lidiar. Así también se promueve que cada estudiante participante del curso asuma el rol de líder del proyecto y experimente, a través de Cadxela GSD, los problemas y retos del GSD relacionados con este tópico.

Por otro lado, al inicio de cada proyecto el equipo recibe un conjunto de recomendaciones propuestas por Cohn (2010), con relación al desarrollo de proyectos de GSD con la aplicación de *Scrum* como metodología de desarrollo. Estas recomendaciones están enfocadas a fortalecer los aspectos de comunicación, trabajo en equipo, colaboración, y planificación del proyecto y se describen de la siguiente manera:

- **Crear coherencia:** La coherencia hace referencia a la unión del equipo y en un entorno de GSD existen diversos factores que la pueden afectar, tales como: el idioma, la cultura, la separación física, y las diferencias de zona horaria. Por tal motivo se sugiere realizar una reunión formal entre todos los miembros del equipo para que éstos se conozcan y compartan ideas y opiniones. Esto mejorará la cohesión y confianza entre los integrantes.
- **Realizar la planeación del proyecto:** Los planes son útiles para determinar si un proyecto está en camino de entregar la funcionalidad que los usuarios necesitan y esperan. En este contexto, se sugiere que después de realizar una reunión para generar coherencia, se realice la planeación del proyecto.

De esta manera, la Figura 3.14 muestra que el equipo de estudiantes debe desarrollar el proyecto de software de la siguiente manera:

1. Realizar un plan de liberación.
2. Realizar la planeación de la iteración.
3. Desarrollar los entregables de la iteración.
4. Repetir paso 2 hasta que termine el proyecto.

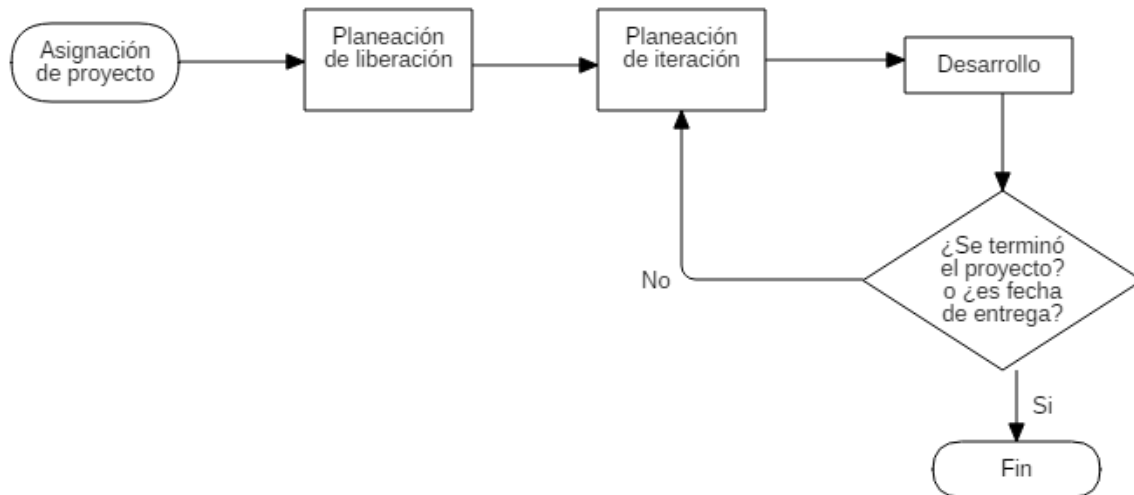


Figura 3.14. Proceso para el desarrollo de los proyectos

3.2.3.1. Desarrollo

Una vez que un proyecto ha sido asignado a un equipo, el estudiante con el rol de líder del proyecto recibe una notificación a través de Cadxela GSD que le informa sobre el proyecto que le fue asignado, además de las recomendaciones mencionadas anteriormente, para que inicie con la planificación del proyecto. En este sentido, en adelante se definirán las actividades que se realizan para el seguimiento y control de los proyectos a través de Cadxela GSD.

3.2.3.1.1. Plan de liberación

Cuando un proyecto es asignado a un equipo, es responsabilidad del estudiante con el rol de líder del proyecto iniciar su planificación. Así, el equipo debe, primeramente, realizar un plan de liberación que permita obtener una visión clara de que se podrá obtener como producto final. En este contexto, Cadxela GSD permite al equipo de trabajo llevar a cabo la planificación de la liberación, tomando como base los lineamientos sugeridos por Cohn (2005) que se resumen en la Figura 3.15.

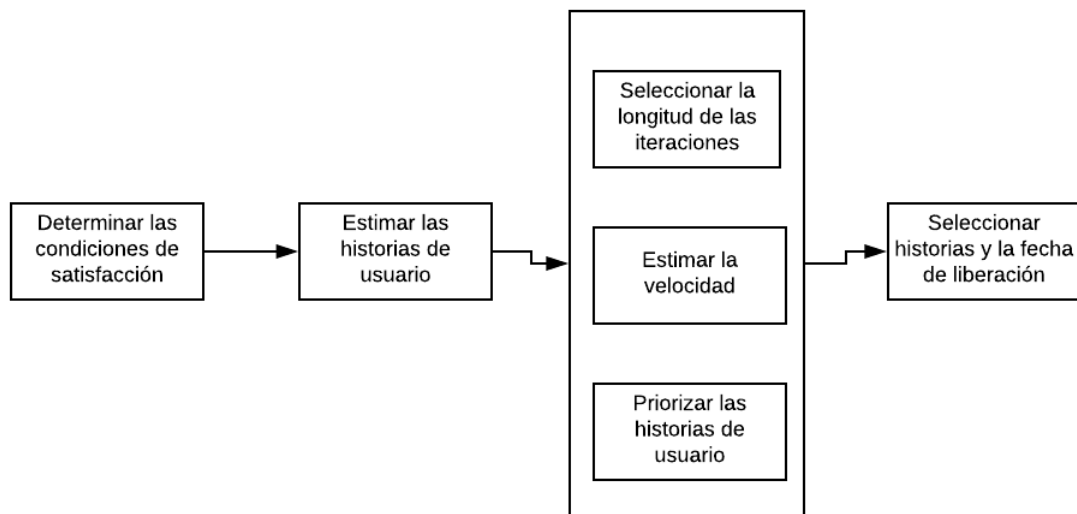


Figura 3.15. Proceso de planeación de la liberación (Cohn, 2005)

Así, el líder del proyecto tiene acceso a un conjunto de pantallas que lo guían en el proceso de creación del plan y que se describen a continuación:

- **Determinar las condiciones de satisfacción:** Antes de iniciar con el plan, el equipo de trabajo junto con el profesor debe determinar aquellos criterios que permitirán evaluar el éxito del proyecto. Esta evaluación se realiza a través de una reunión formal entre los integrantes del equipo y el profesor, en donde se deben dejar en claro cuáles serán aquellos factores de éxito del proyecto.
- **Estimar las historias de usuario:** El equipo de trabajo junto con el profesor debe estimar cada una de las historias de usuario que podrán ser implementadas en el producto final. En este contexto, la estimación se realiza en puntos de historia que representan una unidad de medida utilizada para expresar el tamaño total de una historia de usuario, característica, u otra pieza de trabajo. Los valores asignados son relativos, es decir, una historia a la que se le asigna un 2 debe ser el doble de una historia a la que se le asigna un 1. También debe ser dos tercios de una historia que se estima en 3 puntos de historia. Una estimación del punto de la historia es una mezcla de la cantidad de esfuerzo involucrado en el desarrollo de la característica, la complejidad de su desarrollo, el riesgo inherente en ella, entre otros.
- **Seleccionar la longitud de las iteraciones:** La mayoría de los equipos ágiles trabajan con iteraciones de una a cuatro semanas de longitud. No obstante, es importante elegir una longitud de iteración apropiada.
- **Estimar la velocidad:** Si los integrantes del equipo tienen experiencia trabajando juntos, la mejor opción es establecer la velocidad exhibida más recientemente por el equipo. En caso contrario, es posible utilizar técnicas como la estimación basada en datos históricos, la ejecución de una iteración para obtener valores de velocidad, o hacer un pronóstico de velocidad.
- **Priorizar las historias de usuario:** Generalmente los proyectos tienen muy poco tiempo o están compuesto de muchas características, lo cual dificulta que se desarrollen todas las

características en cierta cantidad de tiempo. Debido a lo anterior, el propietario del producto (e.g., profesor) debe priorizar las características que deben desarrollarse.

- Seleccionar las historias de usuario y la fecha de liberación: Hasta este punto, ya se han obtenido la longitud de las iteraciones, y la velocidad del equipo por iteración, entonces, es momento de realizar un plan de liberación que cumpla las condiciones de satisfacción del proyecto. Dicho plan debe contener las historias de usuario que serán liberadas y su fecha de entrega.

Lo anterior se puede realizar en Cadxela GSD a través de un conjunto de pasos que guían al estudiante, con el rol de líder del proyecto, para crear un plan (véase Figura 3.16).

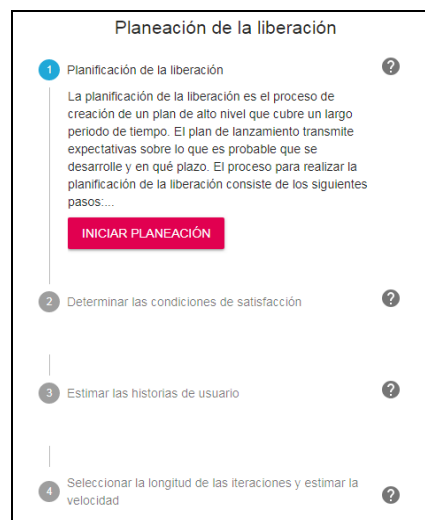


Figura 3.16. Pasos sugeridos por Cadxela GSD para la creación de plan de liberación

En este sentido, el estudiante debe organizar diferentes reuniones para crear el plan de liberación. Por lo tanto, debe crear una reunión para determinar las condiciones de satisfacción del proyecto, la cual involucrará a todos los miembros del equipo de trabajo y al profesor. Las reuniones proporcionan de manera general las siguientes características: un servicio de mensajería instantánea, un servicio de videollamada, y un panel de trabajo para actividades específicas como la estimación de las historias de usuario, y la división de éstas en tareas y su asignación a los participantes, la Fig. 3.17 muestra cómo se realiza una reunión en Cadxela GSD.

Posteriormente, los estudiantes junto con el profesor deben realizar la estimación de las historias de usuario haciendo uso de la técnica de estimación de póker. La estimación de póker combina la opinión de expertos, la analogía (i.e., el estimador compara la historia estimada con una o más historias, si la historia es el doble del tamaño se le da una estimación el doble de grande) y la desagregación (i.e., dividir historias en pequeñas tareas fáciles de estimar) en un enfoque agradable para realizar estimaciones rápidas y confiables. Para llevar a cabo esta estimación, es necesario realizar una reunión que involucre a todos los integrantes del equipo de desarrollo. Al iniciar la reunión, cada integrante del equipo dispone de un mazo de cartas las cuales tienen escritas un valor válido de estimación (e.g., 1, 2, 3, 5, 8). Después, el moderador (i.e., el estudiante que desempeña el rol de líder del proyecto) selecciona una historia de usuario a estimar y la lee a los participantes, y éstos deben seleccionar una carta que represente el peso de la historia que ellos estiman y colocan

sobre la mesa. Cuando todos han seleccionado un valor de estimación, se muestran simultáneamente los valores estimados por todos los participantes. Es muy probable que en este punto las estimaciones difieran significativamente y los participantes deban discutir sobre los valores asignados. Si fuese necesario, el moderador puede repetir el proceso para llegar a un acuerdo sobre el valor que será asignado a la historia de usuario (Cohn, 2005).

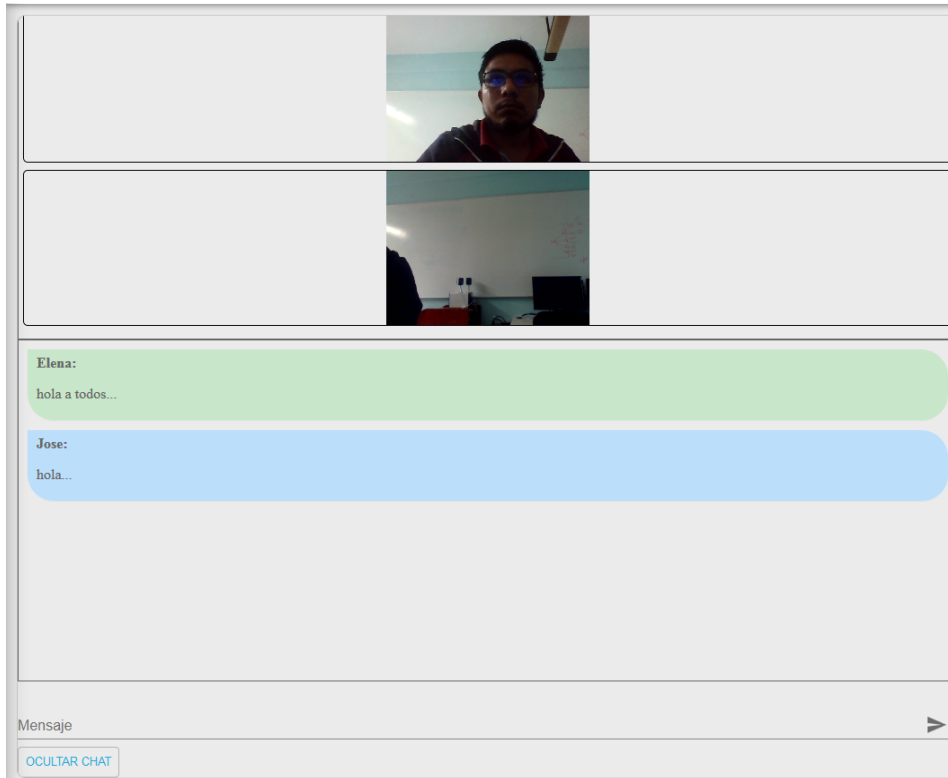


Figura 3.17. Reunión del equipo a través de Cadxela GSD

Esta estimación se realiza en Cadxela GSD a través de una reunión que involucra al equipo de trabajo y al profesor. Inicialmente, el moderador selecciona una historia de usuario a estimar que será enviada a todos los participantes para poder realizar su estimación. Después, cada uno de los participantes debe seleccionar un valor de estimación haciendo *clic* sobre una de las cartas que aparecen en la parte inferior de la pantalla, la cuál será actualizada automáticamente en las sesiones del resto de integrantes (véase Figura 3.18). Cuando el moderador recibe todas las estimaciones, gira todas las cartas y éstas son mostradas al mismo tiempo a todos los participantes. Además, Cadxela GSD provee de una herramienta de videollamada y mensajería instantánea para fomentar la discusión sobre las estimaciones hechas. Cuando se llegue a un acuerdo sobre el valor estimado, el moderador debe asignar dicha estimación a la historia y repetir el proceso con otra historia de usuario.

Por otro lado, la longitud de las iteraciones es establecida por el estudiante con el rol de líder del proyecto, siguiendo las recomendaciones establecidas por Cohn (2005), las cuales sugieren que se deben considerar los valores históricos de proyectos anteriores siempre, y cuando se cuente con ellos, así como la cantidad de incertidumbre del proyecto, la facilidad para obtener mayor realimentación del proyecto durante la iteración (i.e., cantidad, frecuencia, y tiempo de interacciones

entre el equipo de trabajo), la posibilidad de cambios en las prioridades del proyecto, y los costos de la iteración.

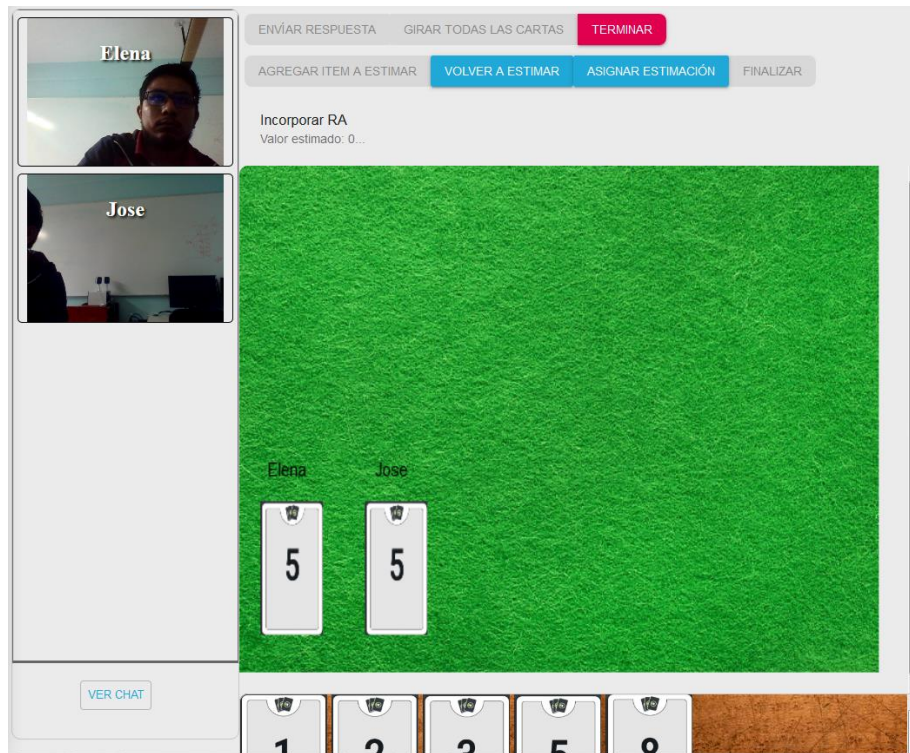


Figura 3.18. Estimación de historias de usuario a través de Cadxela GSD

En el caso de la estimación de la velocidad, el equipo puede realizarla a través de una simulación que les permita obtener la cantidad de puntos de historia que podrán cubrir en una iteración. En este sentido, las actividades que es necesario realizar son las siguientes:

1. Estimar la cantidad de horas que cada miembro del equipo de trabajo dedicará al proyecto: Esta es una actividad personal que cada integrante realizará para agregar un valor numérico a su perfil de Cadxela GSD.
2. Determinar la cantidad total de horas que se dedicarán al proyecto: Cuando todos los integrantes del equipo hayan ingresado el tiempo dedicado al proyecto, el sistema habilitará al líder del proyecto la opción para realizar el cálculo total de horas. El total de horas se calcula sumando la cantidad de horas dedicadas por cada uno de los integrantes durante cada iteración.
3. Seleccionar una historia de usuario y dividirla en tareas: El estudiante con el rol de líder del proyecto debe organizar una reunión para que los integrantes del equipo de trabajo realicen la división de historias de usuario en tareas y actualicen el tiempo estimado para su realización (véase Figura 3.19). Este paso se debe repetir hasta que se identifiquen suficientes tareas que cubran las horas estimadas para la iteración.

Finalmente, el profesor debe priorizar las historias de usuario y los estudiantes deben seleccionar aquellas que serán desarrolladas con base en la estimación que han realizado y, de esta

manera, obtener un plan de liberación. Cabe destacar que el profesor podrá visualizar en todo momento el estado de la planeación que realiza cada uno de los equipos, como se muestra en la Figura 3.20.

Historia	Puntos de historia estimado	Horas estimadas
Incorporar RA	5	24
Total	5	24

Figura 3.19. Estimación de velocidad a través de Cadxela GSD

Figura 3.20. El profesor puede ver el progreso del equipo

3.2.3.1.2. Planeación de la iteración.

Una vez que el equipo ha terminado de planificar la liberación, ya cuenta con datos sobre la cantidad de iteraciones que deberán realizar, la longitud de cada una estas iteraciones (en semanas), el tiempo en horas de cada iteración, los puntos de historia que se realizarán en la iteración, y el total

de puntos de historia que podrán ser desarrollados. A continuación, deben de llevar a cabo la planeación de la primera iteración. Para ello, el estudiante con el rol de líder de proyectos deberá programar una reunión donde se realice la planeación de la iteración con todos los integrantes del equipo de trabajo.

La reunión de planeación es similar a la reunión de estimación de la velocidad, con la diferencia de que en este punto las historias de usuario ya han sido priorizadas por el profesor y además, ya sólo se muestran aquellas historias de usuario que pueden ser completadas durante el proyecto. Al finalizar esta reunión, el equipo de trabajo tiene ya las tareas que deberán desarrollar durante la iteración y quienes serán los responsables de cada una de las tareas.

3.2.3.1.3. Entregables de la iteración

Con la planificación de las iteraciones, los estudiantes tienen acceso a un tablero de tareas que permite al equipo organizar el trabajo y ver el progreso del desarrollo de la iteración. En este tablero, los estudiantes pueden ver las tareas que les fueron asignadas, así como las tareas que realizan sus compañeros. Cada tarea contiene una breve descripción de ésta, el tiempo requerido en horas para su realización, y el responsable de hacerla. Por otro lado, el tablero está compuesto de cuatro columnas: “*Por hacer*”, “*Haciendo*”, “*Verificando*”, y “*Terminadas*” que muestran el progreso de cada una de las tareas. Inicialmente, todas las tareas son colocadas en la columna “*por hacer*”, y es responsabilidad de cada estudiante mover su tarea a la columna correspondiente de acuerdo con los siguientes criterios:

- **Por hacer:** Esta columna muestran aquellas tareas que se han identificado como necesarias para realizar una historia de usuario, pero que aún no se ha trabajado sobre de ellas.
- **Haciendo:** Esta columna muestra aquellas tareas que ya están siendo ejecutadas por el responsable asignado.
- **Verificando:** Esta columna muestra aquellas tareas que ya han sido ejecutadas y están en proceso de verificación.
- **Terminadas:** Esta columna muestra aquellas tareas que ya han sido terminadas y verificadas.

Cuando una tarea es colocada en la columna de “*Terminadas*”, se actualiza el progreso de la iteración. Además, solamente el responsable de la tarea puede actualizar el estado de ésta. Finalmente, la Figura 3.21 muestra el tablero de tareas al que los estudiantes tienen acceso, en donde se puede apreciar el progreso de la iteración, así como el estado de las diferentes tareas que forman parte de éste.



Figura 3.21. Tablero de tareas en Cadxela GSD

3.2.4. Monitorización y control del proyecto

Con el objetivo de apoyar al estudiante con el rol de líder de proyectos en la monitorización y control del proyecto, Cadxela GSD proporciona al estudiante información relacionada con el progreso de cada una de las iteraciones, como se mostró anteriormente con el tablero de tareas. Aunado a lo anterior, la plataforma proporciona también una gráfica del progreso de la liberación que muestra la cantidad de trabajo realizado y el trabajo restante para terminar el proyecto (véase Figura 3.22). En este contexto, el eje vertical muestra la cantidad de puntos de historia que fueron estimados por el equipo de trabajo, mientras que el eje horizontal muestra el progreso que se tiene en el proyecto a través de una línea de tiempo que cubre desde su inicio hasta su terminación.

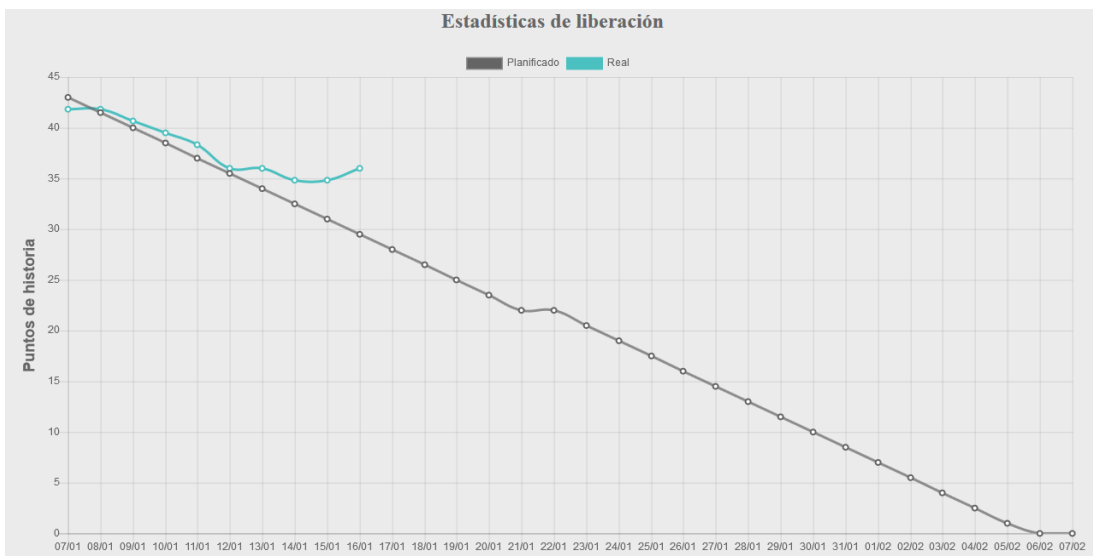


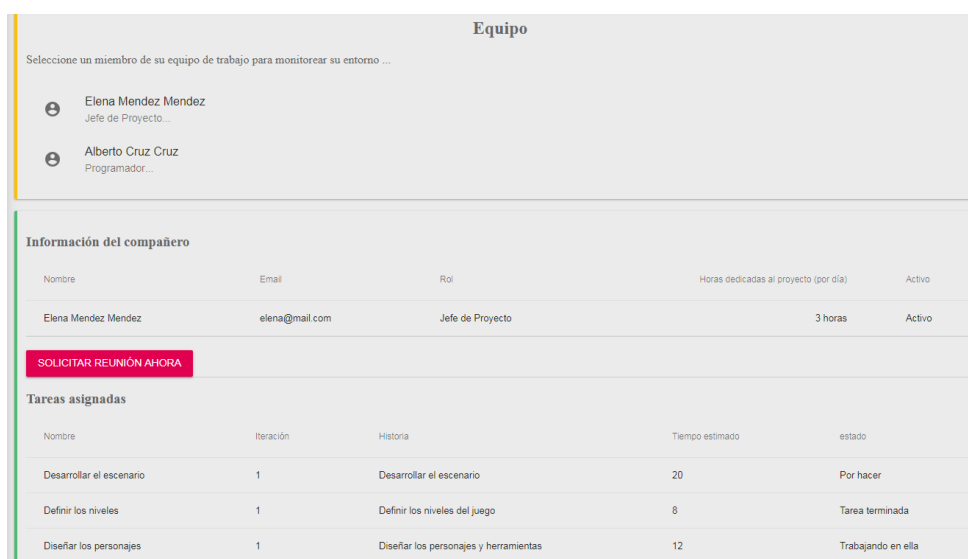
Figura 3.22. Estadísticas del progreso del proyecto

3.2.5. Comunicación formal e informal

Otro aspecto importante que ha sido integrado en Cadxela GSD y que sirve de apoyo en el desarrollo del proyecto, es el concepto de las esferas de trabajo colaborativas (CWS). En este

sentido, una CWS permite a los estudiantes identificar el momento oportuno para la comunicación y colaboración informal. Para ello, el estudiante debe acceder a la pantalla de comunicación informal dentro de Cadxela GSD. Una vez que se haya seleccionado esta opción, el sistema le mostrará una pantalla donde debe seleccionar a un miembro de su equipo de trabajo y se le mostrará automáticamente información relevante acerca de su compañero como, por ejemplo, las horas dedicadas al proyecto, las tareas asignadas a su compañero, el estado de cada una de esas tareas, y si está o no disponible para realizar una comunicación informal. En caso de estar disponible, el sistema mostrará al estudiante la opción para solicitar una comunicación informal (véase Figura 3.23).

Por otro lado, el estudiante con el rol de líder de proyectos tiene acceso a las estadísticas de interacciones entre miembros de su equipo de trabajo (e.g., número de interacciones entre cada uno de los integrantes del equipo, hora en la que se realizan estas interacciones, entre otras). Los estudiantes pueden realizar una comunicación formal a través de la programación de reuniones entre los integrantes del equipo. Estas reuniones pueden ser programadas por el estudiante con el rol de líder de proyectos accediendo a la opción reuniones dentro de Cadxela GSD. Una vez que esta opción sea seleccionada, el estudiante accederá a una pantalla donde se muestra una lista de las reuniones realizadas y por hacer para su equipo de trabajo. Además, podrá programar una nueva reunión (véase Figura 3.24).



Equipo

Seleccione un miembro de su equipo de trabajo para monitorear su entorno ...

- Elena Mendez Mendez
Jefe de Proyecto...
- Alberto Cruz Cruz
Programador...

Información del compañero

Nombre	Email	Rol	Horas dedicadas al proyecto (por día)	Activo
Elena Mendez Mendez	elena@mail.com	Jefe de Proyecto	3 horas	Activo

SOLICITAR REUNIÓN AHORA

Tareas asignadas

Nombre	Iteración	Historia	Tiempo estimado	estado
Desarrollar el escenario	1	Desarrollar el escenario	20	Por hacer
Definir los niveles	1	Definir los niveles del juego	8	Tarea terminada
Diseñar los personajes	1	Diseñar los personajes y herramientas	12	Trabajando en ella

Figura 3.23. Concepto de CWS en Cadxela GSD



NUEVA REUNIÓN

Reunión para: Planificación de la iteración #1... Terminado... ^

ENTRAR

Estimar la velocidad del equipo...	En progreso...	▼
Conocimiento...	En progreso...	▼
Reunión para realizar estimación de tareas...	Terminado...	▼
Reunión para: Planificación de la iteración #1...	Terminado...	▼

Figura 3.24. Comunicación formal en Cadxela GSD

3.2.6. Evaluación de los proyectos

Finalmente, el profesor podrá evaluar el rendimiento de los equipos de trabajo y de los estudiantes en particular. En este sentido, Cadxela GSD proporciona al profesor estadísticas del progreso de cada uno de los equipos y estudiantes tales como:

- La gráfica del progreso de la liberación, que le muestra el progreso total del equipo y el trabajo restante para la terminación del proyecto.
- Información del trabajo realizado por el equipo a través del tablero de tareas de la iteración.
- La información del estado actual de cada estudiante (e.g., tareas asignadas y su progreso actual, número de interacciones informales en el equipo), como se aprecia en la Figura 3.25.
- Estadísticas relacionadas con la comunicación formal e informal (e.g., número de reuniones formales e informales, horario más común para la comunicación formal e informal, entre otras) acerca de la información relacionada con las reuniones formales del equipo de trabajo (véase Figura 3.26).

Información del estudiante				
Nombre	Email	Rol	Horas dedicadas al proyecto (por día)	
María del Rosario Pérez Méndez	maria@mail.com	Analista	3 horas	

Tareas asignadas				
Nombre	Iteración	Historia	Tiempo estimado	estado
asdasdas	1	El usuario podrá visualizar todos los pedidos realizados con una descripción detallada	35	Por hacer
dsfdsdf	1	El usuario podrá seleccionar los productos de mayor interés	25	Por hacer

Participación en reuniones informales				
Participación	Con	Estado	Inicio	Fin
Iniciador	Mesut Ozil	Aceptada		2019-01-25T16:01:04.470Z
Iniciador	Mesut Ozil	Aceptada		2019-01-25T15:39:58.815Z
Iniciador	Mesut Ozil	Aceptada	2019-01-25T15:37:21.502Z	
Iniciador	Mesut Ozil	Rechazada		

Figura 3.25. Información por estudiante en Cadxela GSD

Lo anterior permite que el profesor realice evaluaciones con respecto a la gestión del proyecto realizada por el equipo, el desempeño del equipo para la realización del proyecto, aspectos de comunicación en el entorno de GSD, etc.



Figura 3.26. Información de reuniones formales del equipo

Una vez que el marco conceptual y la plataforma Cadxela GSD han sido descritos, el siguiente capítulo de la tesis relata el caso de estudio diseñado para evaluar el grado en que se contribuye al desarrollo de las habilidades suaves, necesarias en entornos de GSD.

4. Resultados experimentales

Con el objetivo de validar el marco de referencia y la plataforma propuestos en esta tesis, se diseñó y condujo un caso de estudio que contó con la participación de cinco universidades mexicanas, entre las cuales están la Universidad Tecnológica de la Mixteca (UTM) ubicada en la Ciudad de Huajuapán de León, Oaxaca y cuatro universidades más (U1 ubicada en Oaxaca, U2 ubicada en Puebla, U3 ubicada en Baja California Sur, y U4 ubicada en Ciudad de México) cuyos nombres son omitidos por cuestiones de confidencialidad. En las siguientes secciones se describe a detalle el diseño del caso de estudio, su ejecución, y los resultados obtenidos.

4.1. Caso de estudio

4.1.1. Participantes

Participaron un total de veinte estudiantes y cinco profesores de las universidades U1, U2, U3, U4, y UTM mencionadas anteriormente, las cuales cumplieron con las características principales para poder llevar a cabo la validación de la propuesta: están ubicadas en diferentes zonas geográficas (Oaxaca, Puebla, Ciudad de México, y Baja California Sur); presentan diferencias de culturas, principalmente con relación a la forma de trabajo de sus profesores y estudiantes, ya que la universidad U1 y U4 trabajan con la modalidad de clases a distancia por lo que sus estudiantes tienen una forma de trabajo diferente a la de sus compañeros, además las diferencias en la forma de vida y lenguaje entre la zona norte y sur de México permiten que se aumente las diferencias culturales entre los participantes; y se ubican en diferente zona horaria, concretamente entre la universidad U3 y el resto de los participantes existe una diferencia temporal de 2 horas. Así, los estudiantes pudieron experimentar las distancias geográficas, temporal, y de cultura en el desarrollo de proyectos de software.

Cada universidad asignó un profesor como responsable de la administración del curso dentro de su institución, al cual le fue asignado un equipo conformado por estudiantes de las diferentes universidades para que éste supervisara su trabajo y desempeño en el proyecto que les fuera asignado. En total se contó con la participación de cinco equipos conformados por cuatro estudiantes que cursaban una carrera relacionada con la IS en cada universidad. El objetivo fue que los estudiantes formaran equipos de trabajo con sus similares de otras universidades para desarrollar pequeños proyectos de GSD.

4.1.2. Instrumentos

El caso de estudio pretendió evaluar la eficiencia del marco de referencia propuesto en términos de aprovechamiento académico de los estudiantes con respecto al GSD, mejora en habilidades de trabajo en equipo y comunicación, rendimiento en el desarrollo de los proyectos, la

motivación y satisfacción de los estudiantes con el enfoque establecido, y la aceptación por parte de los profesores participantes.

Con estos objetivos en mente, se inició por impartir a los estudiantes de cada una de las universidades los conceptos teóricos sugeridos por la definición del alcance del marco de referencia (véase sección 3.1.4.1). Lo anterior permitió que los estudiantes recibieran conceptos teóricos sobre el GSD para que, de esta manera, pudieran ponerlos en práctica en los proyectos que les fueron asignados, como también sugiere el marco de referencia.

Una vez que los estudiantes fueron capacitados en los conceptos del GSD, se formaron los equipos de trabajo. Cada uno de los equipos contó con la participación de un estudiante de las diferentes universidades, lo que permitió la integración de equipos en donde los participantes trabajarían juntos por primera vez, con el objetivo de experimentar el trabajo en equipo, comunicación, y colaboración entre personas con diferente cultura y forma de trabajo. Los equipos fueron organizados de tal manera que se pudiera garantizar la experimentación sobre el GSD, ya que entre cada una de las universidades que participan, existen marcadas distancias tanto geográficas como temporales.

Por otro lado, se realizó una pre-evaluación de manera individual al final del curso teórico con relación al conocimiento sobre el GSD, con el fin de determinar el nivel de conocimiento de los estudiantes antes de la experimentación práctica con los proyectos. De manera similar, se realizó una post-evaluación al final del curso con el objetivo de determinar la eficacia del marco de referencia para mejorar el conocimiento sobre el GSD. En este sentido, los instrumentos de evaluación fueron diseñados siguiendo los lineamientos de los cuestionarios mixtos, que consideran la incorporación de preguntas tanto cerradas como abiertas. Las preguntas cerradas son aquellas en donde se espera una respuesta breve, específica, y delimitada. En particular, se esperan respuestas dicotómicas (e.g., sí o no) o respuestas de opción múltiple. Las preguntas abiertas, por otro lado, son aquellas que no delimitan a la persona que responde el cuestionario a un conjunto de respuestas pre establecidas, sino que ésta es libre de redactarla (Rojas, 1998). Es así como los cuestionarios de evaluación de conocimientos aplicados a los estudiantes estuvieron conformados por tres preguntas abiertas relacionadas con la definición del GSD, sus beneficios, y a que se le denomina equipo virtual. Así mismo, se incluyeron 17 preguntas cerradas de opción múltiple para cuestionar a los estudiantes sobre el problema de las tres distancias, aspectos relacionados a la comunicación y colaboración en entornos de GSD, técnicas ágiles implementadas en Cadxela GSD, aspectos de la gestión del conocimiento, y diversidad cultural. Finalmente, para obtener el grado de aprovechamiento académico se comparó el promedio de respuestas obtenidas por todos los estudiantes en las dos evaluaciones. A modo de ilustración, la Tabla 11 muestra el cuestionario de evaluación aplicado a los estudiantes.

Tabla 11. Cuestionario de evaluación de conocimientos del GSD.

#	Pregunta			
P1	Define de manera breve y concisa el enfoque de GSD.			
P2	Menciona al menos tres beneficios de este enfoque.			
P3	Define de manera breve y concisa a qué se denomina “ <i>equipo virtual</i> ” en el GSD.			
P4	El concepto conocido como las “3 C’s” hace referencia a problemas en: <table style="width: 100%; border: none;"> <tr> <td style="width: 33%; vertical-align: top;">a) Colaboración, coordinación, y comunicación</td> <td style="width: 33%; vertical-align: top;">b) Comunicación, coordinación, y control</td> <td style="width: 33%; vertical-align: top;">c) Control, colaboración, y comunicación</td> </tr> </table>	a) Colaboración, coordinación, y comunicación	b) Comunicación, coordinación, y control	c) Control, colaboración, y comunicación
a) Colaboración, coordinación, y comunicación	b) Comunicación, coordinación, y control	c) Control, colaboración, y comunicación		

#	Pregunta
P5	<p>Cuando se dice “<i>Es la medida de esfuerzo que un individuo necesita realizar para visitar otro punto, alejado del primero</i>”, se refiere a la definición de:</p> <p>a) Distancia temporal b) Distancia cultural c) Distancia geográfica</p>
P6	<p>Cuando se dice “<i>Es la medida de la deslocalización en tiempo, experimentada por dos individuos que desean interactuar</i>”, se refiere a la definición de:</p> <p>a) Distancia temporal b) Distancia cultural c) Distancia geográfica</p>
P7	<p>Cuando se dice “<i>Es la medida en que un individuo comprende las costumbres (símbolos, normas, y valores sociales) y cultura de otro individuo</i>”, se refiere a la definición de:</p> <p>a) Distancia temporal b) Distancia cultural c) Distancia geográfica</p>
P8	<p>¿Cuál es el tipo de comunicación que puede darse en cualquier momento durante el desarrollo de un proyecto?</p> <p>a) Formal b) Informal c) Semi-formal</p>
P9	<p>Problemas tales como retrasos en la comunicación, dificultad para establecer reuniones, y dependencia de infraestructura técnica para la comunicación son originados regularmente por la:</p> <p>a) Distancia cultural b) Distancia geográfica c) Distancia temporal</p>
P10	<p>Problemas tales como malentendidos en la comunicación derivados del lenguaje, problemas en la gestión de compromisos, acuerdos, y adaptación a las estructuras normativas son originados regularmente por la:</p> <p>a) Distancia temporal b) Distancia cultural c) Distancia geográfica</p>
P11	<p>En la metodología <i>Scrum</i> se realiza un tipo de planeación que se lleva a cabo durante el inicio del proyecto y es el proceso a través del cual se crea un plan de alto nivel que cubre un largo periodo de tiempo, durante el cual se desarrollará un proyecto de software. En concreto, éste se conoce como plan de:</p> <p>a) Liberación b) Producto c) Iteración</p>
P12	<p>En la metodología <i>Scrum</i> se realiza un tipo de planeación en donde se habla de las tareas que serán necesarias para transformar una solicitud de ciertas características requeridas en un software funcional y probado que será entregado en el siguiente incremento. En concreto, éste se conoce como plan:</p> <p>a) Diario b) De producto c) De iteración</p>
P13	<p>En la metodología <i>Scrum</i> se realiza un tipo de planeación en donde los equipos limitan el horizonte de planificación para no estar más lejos que el día siguiente, cuando se reúnan de nuevo. En concreto, éste se conoce como:</p> <p>a) Plan de liberación b) Plan diario c) Plan de iteración</p>
P14	<p>En este enfoque se pretende relatar los requisitos con un lenguaje coloquial, en lugar de escribirlos de manera formal:</p> <p>a) Temas b) Historias de usuario c) Epopeyas</p>
P15	<p>Representa una unidad de medida utilizada para expresar el tamaño total de una historia de usuario, característica, u otra pieza de trabajo en <i>Scrum</i>:</p> <p>a) Tiempo ideal b) Puntos de historia c) Esfuerzo</p>
P16	<p>Es una técnica de estimación utilizada en <i>Scrum</i> en donde cada estimador dispone de un mazo de cartas con una escala de evaluación válidas y a cada historia de usuario se le asigna un valor estimado contenido en éstas:</p> <p>a) Opinión de expertos b) Analogía c) Estimación de póker</p>
P17	<p>Es un método que simplifica el proceso de compartir, distribuir, crear, capturar y comprender el conocimiento de una empresa, y ayuda a garantizar el desarrollo de un producto que tenga las características adecuadas y un nivel apropiado de calidad:</p>

#	Pregunta		
	a) Distribución del conocimiento	b) Gestión del conocimiento	c) Colaboración
P18	Es un enfoque utilizado en el GSD para la gestión del conocimiento:		
	a) El enfoque de la escuela de ingeniería	b) Enfoque basado en las personas	c) Enfoque basado en la tecnología
P19	En los entornos de GSD, la diversidad cultural se manifiesta en tres dimensiones principales:		
	a) Lugar de trabajo, sociedad, y trabajo	b) Lugar de trabajo, trabajadores, y trabajo	c) Trabajadores, sociedad, y trabajo
P20	Los problemas originados de los diferentes niveles del contexto sociocultural que van desde la sociedad nacional a la regional, organizacional, profesional y de equipo afectan directamente a:		
	a) Lugar de trabajo	b) Trabajadores	c) Trabajo

Así mismo, todos los estudiantes respondieron una encuesta para evaluar su motivación, satisfacción, y experiencia de aprendizaje durante la ejecución del caso de estudio y la efectividad de la propuesta en relación con el aprendizaje del GSD y los lineamientos establecidos por el marco de referencia. Dicha encuesta fue diseñada empleando una escala tipo Likert de 4 puntos con las siguientes alternativas de respuesta: “Totalmente en desacuerdo” (TD = 1), “En desacuerdo” (D = 2), “De acuerdo” (A = 3), y “Totalmente de acuerdo” (TA = 4) que fueron convertidas a un valor numérico para obtener una medida cuantitativa. Por lo tanto, se determinó el número total de respuestas y se calculó la media (M) para todas las respuestas de los estudiantes. La escala de 4 puntos se seleccionó intencionalmente para eliminar el factor de indecisión entre los estudiantes que utilizan “elementos socialmente deseables” como medida semi-forzada y obligar a que eligieran un lado (positivo o negativo) (Weisberg, 2009).

Aunado a lo anterior, una encuesta dirigida a los profesores fue diseñada con la finalidad de obtener el grado de aceptación de Cadxela GSD como herramienta de apoyo en la enseñanza del GSD. La encuesta fue diseñada tomando en cuenta las recomendaciones del Modelo de Aceptación de la Tecnología (TAM, por sus siglas en inglés) y del Modelo de la Teoría Unificada de Aceptación del Uso de la Tecnología (UTAUT, por sus siglas en inglés). De acuerdo con Al-Qeisi (2009), el TAM es un modelo que tiene por objetivo evaluar el nivel de aceptación de los usuarios con respecto a una tecnología basándose en dos variables principales:

- Utilidad percibida (UP): Hace referencia al grado en el cual una persona considera que el uso de un sistema en particular le ayudará a mejorar el rendimiento en su trabajo.
- Facilidad de uso percibida (FU): Pretende evaluar el grado en el cual una persona piensa que el utilizar un sistema en particular le requerirá el menor esfuerzo posible.

Además, en el modelo se considera que existen variables externas que afectan directamente a la utilidad percibida y a la facilidad de uso percibida. Estas variables externas hacen alusión a las características del diseño del sistema, el entrenamiento de los usuarios, el rendimiento computacional, la participación de los usuarios en el diseño, y la naturaleza del proceso de implementación. Por otro lado, el UTAUT es un modelo presentado por Venkatesh et. al. (como se citó en Al-Qeisi, 2009) que propone una visión unificada para evaluar la aceptación de la tecnología. De esta manera, los autores identificaron cuatro constructos que impactan directamente en la

aceptación de la tecnología por parte de los usuarios. Es importante agregar que el modelo considera que existen variables modificadoras que afectan directamente a dichos constructos, las cuales son: la edad, el género, y la experiencia de las personas. Por lo tanto, los constructos considerados son:

- Expectativa de rendimiento (ER): Es el grado en el cual una persona cree que el utilizar un sistema en particular le permitirá mejorar su rendimiento laboral.
- Expectativa de esfuerzo (EE): Es el grado de facilidad de uso de un sistema en particular.
- Influencia social: Es el grado en el cual una persona percibe que otras personas importantes para ella piensan que se debería utilizar el sistema.
- Condiciones facilitadoras (CF): Es el grado en el cual una persona cree que existe una infraestructura técnica y organizativa que le brindará ayuda durante el uso del sistema.

Considerando lo anterior, la Tabla 12 muestra el instrumento diseñado para determinar el grado de aceptación de los profesores sobre Cadxela GSD. Es importante mencionar que no se consideró al constructo de influencia social, puesto que Cadxela GSD no se ha expuesto a más profesores, y los que hicieron uso de ella no tienen interacción directa, por lo cual no existe como tal una influencia social. Por último, la encuesta fue evaluada de manera similar a la encuesta de motivación y satisfacción de los estudiantes aplicando una escala tipo Likert de 4 puntos.

Tabla 12. Encuesta aplicada para obtener el grado de aceptación de Cadxela GSD por parte de los profesores.

#	TAM	UTAUT	Ítem
1	UP	ER	Pienso que el uso de Cadxela GSD me ayudó en el proceso de la enseñanza práctica del GSD
2			Pienso que el uso de Cadxela GSD me ayudó a preparar de mejor manera mis clases sobre el GSD
3			Pienso que el uso de Cadxela GSD permitió que mis estudiantes mejoraran sus conocimientos y habilidades relacionadas con el GSD
4			Pienso que Cadxela GSD sería una herramienta útil en mi trabajo docente
5			Pienso que el uso de Cadxela GSD permitió que mis estudiantes estén más motivados por aprender y reforzar los conceptos teóricos del GSD
6	FU	EE	Me resultó fácil organizar equipos y asignarles proyectos a través de Cadxela GSD
7			Me resultó fácil monitorear el progreso de los equipos a través de Cadxela GSD
8			Me resultó fácil comunicarme con mi equipo de estudiantes a través de Cadxela GSD
9			Interactuar con Cadxela GSD no requirió de mucho esfuerzo
10			Tampoco requerí de mucho esfuerzo para aprender a utilizar Cadxela GSD
11	--	CF	Cuando requerí de ayuda para resolver algún problema con Cadxela GSD, conté con instrucciones para ayudarme a hacerlo
12			Cuando necesité de ayuda para resolver algún problema con Cadxela GSD, siempre hubo una persona dispuesta a ayudarme
13			Es más interesante enseñar los conceptos relacionados con el GSD siguiendo el enfoque establecido por Cadxela GSD

Finalmente, el rendimiento de los estudiantes con relación a los proyectos de GSD desarrollados y sus habilidades de trabajo en equipo y comunicación fueron evaluados a través del análisis que los profesores responsables hicieron sobre su trabajo, tanto individual como grupalmente. En este sentido, se hizo uso de la información proporcionada por Cadxela GSD: gráficas, tableros de actividades, porcentajes de reuniones formales e informales, duración de las reuniones, frecuencia de la comunicación, colaboración entre los estudiantes, etc., y el cumplimiento de los planes realizados por los equipos a través de Cadxela GSD.

4.1.3. Desarrollo del caso de estudio

4.1.3.1. Preparación del curso

Para iniciar con el caso de estudio fue necesario que cada universidad participante designara a un profesor responsable para que éste fuera registrado en Cadxela GSD. Así mismo, las universidades designaron a los estudiantes que participarían en el curso, los cuales cursaban los últimos semestres de la Licenciatura en Informática e Ingeniería en Computación y, además, ya habían tomado anteriormente un curso relacionado con la IS por lo que se asumió que poseían conocimiento de lenguajes de programación orientados a objetos y bases de datos. Posteriormente, se definió con los profesores y las autoridades de las universidades un horario y lugar para llevar a cabo el curso teórico que sería impartido a los estudiantes. En este sentido, se acordó con las universidades que el curso sería impartido solamente por un profesor para garantizar así que todos los estudiantes fueran capacitados de la misma manera en los tópicos del GSD. Así, una de las universidades tomó el curso de manera presencial mientras que las otras lo tomaron con la modalidad a distancia.

Una vez que cada universidad designó al profesor y a sus estudiantes, el administrador del sistema registró la información de éstas, de los profesores responsables y del curso en sí (duración, participantes, descripción, etc.) para, de esta manera, iniciar oficialmente con el caso de estudio. En adelante se describe tanto la ejecución del caso de estudio como el desarrollo de los proyectos de GSD.

4.1.3.2. Ejecución del curso teórico

El caso de estudio inició con un curso teórico en donde se capacitó a los estudiantes en los temas sugeridos en el marco de referencia con relación al GSD. La duración del curso fue de 12 días y se impartió un tema por día, combinando conceptos teóricos con una serie de ejercicios y actividades prácticas que se debieron realizar con el objetivo de determinar el nivel de comprensión de cada tema. Concretamente, se abordaron los temas sugeridos en el marco de referencia siguiendo un enfoque de simple a complejo para abordar, inicialmente, la definición del GSD. Este tema se impartió en una clase que tuvo una duración aproximada de 40 minutos. Al final, se les pidió a los estudiantes que realizaran una investigación sobre el tema expuesto y que entregaran un reporte escrito. A modo de ejemplo, la Figura 4.1 muestra un par de imágenes de la clase impartida.

En el siguiente tema se expusieron los beneficios del enfoque de GSD y se trató de concientizar a los estudiantes sobre la importancia del mismo en la industria actual de software. De igual manera, se solicitó a los estudiantes un reporte escrito sobre otros beneficios no mencionados en la clase y ejemplos de empresas reales que estuvieran utilizando el GSD. Al siguiente día se debatió acerca de los hallazgos de los estudiantes, los cuales permitieron que éstos se percataran que el GSD realmente se está implementado en entornos reales de desarrollo.

Posteriormente se abordó el tema de los problemas y retos presentes en el GSD. Con este tema los alumnos se dieron cuenta de que existen aún muchos problemas y retos a resolver para obtener

los beneficios que el GSD plantea y que se habían analizado en la clase anterior. Igualmente, los estudiantes entregaron un reporte acerca de otros problemas que no fueron mencionados en clase.

Durante los siguientes cinco días se impartieron temas relacionados con la implementación de las metodologías ágiles enfocadas a la gestión de proyectos en entornos de GSD. El enfoque seguido para estos temas consistió en proporcionar los conceptos teóricos a los estudiantes y realizar ejercicios prácticos en clase y fuera de clase. Por ejemplo, se recomendó que, para obtener los requisitos de los proyectos, los estudiantes podían aplicar la técnica de juego de roles, la cual es definida por Van Ments (como se citó en Nestel y Tierney, 2007) como: *“Un tipo particular de simulación que enfoca la atención en la interacción de las personas entre sí. Enfatiza las funciones realizadas por diferentes personas en diversas circunstancias”*. Por lo tanto, el juego de roles consiste en platicar con una persona con el fin de imaginar que son ellos mismos u otra persona en una situación particular. Como resultado de esto, la persona que lleva a cabo el rol, así como sus compañeros, aprenden sobre dicha persona, su contexto, y su forma de actuar.

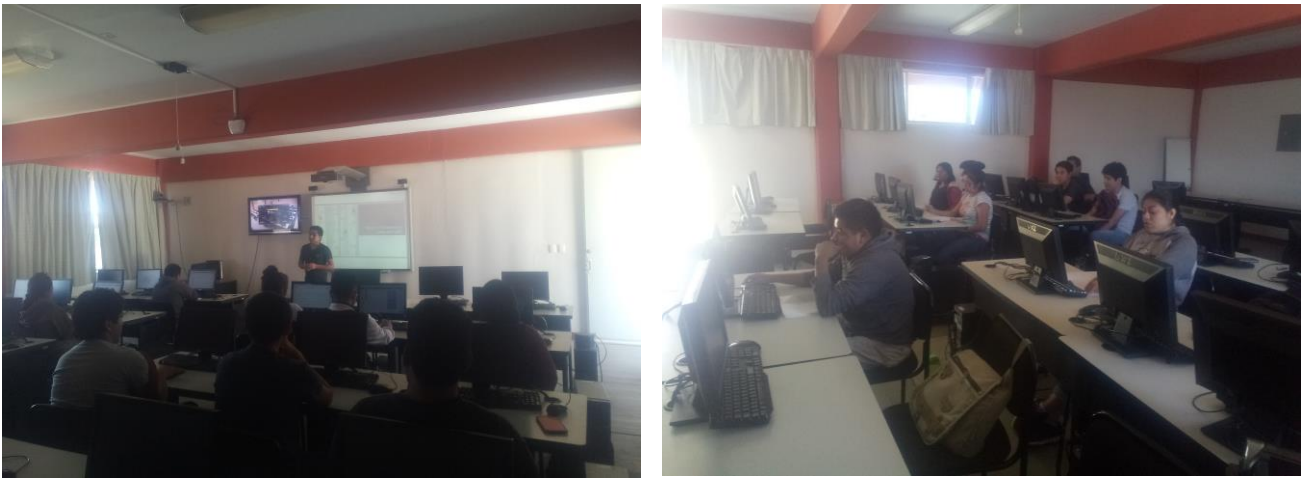


Figura 4.1. Curso teórico acerca del GSD

Con relación a lo anterior, se explicó a los estudiantes en qué consistía la técnica de juego de roles y posteriormente se realizó un ejercicio práctico en donde se les explicó acerca de las historias de usuario y cómo podrían obtenerlas aplicando dicha técnica en los proyectos que les serían asignados.

Para este ejercicio en particular se les planteó a los estudiantes una situación de la vida real sobre la necesidad de un sistema, después ellos deberían de identificar los roles que participan en este escenario, así como la importancia de éstos y qué podrían aportar a los requisitos del sistema. Finalmente, los estudiantes realizaron el ejercicio y obtuvieron una lista de historias de usuario. La Figura 4.2 muestra a los estudiantes realizando el ejercicio.

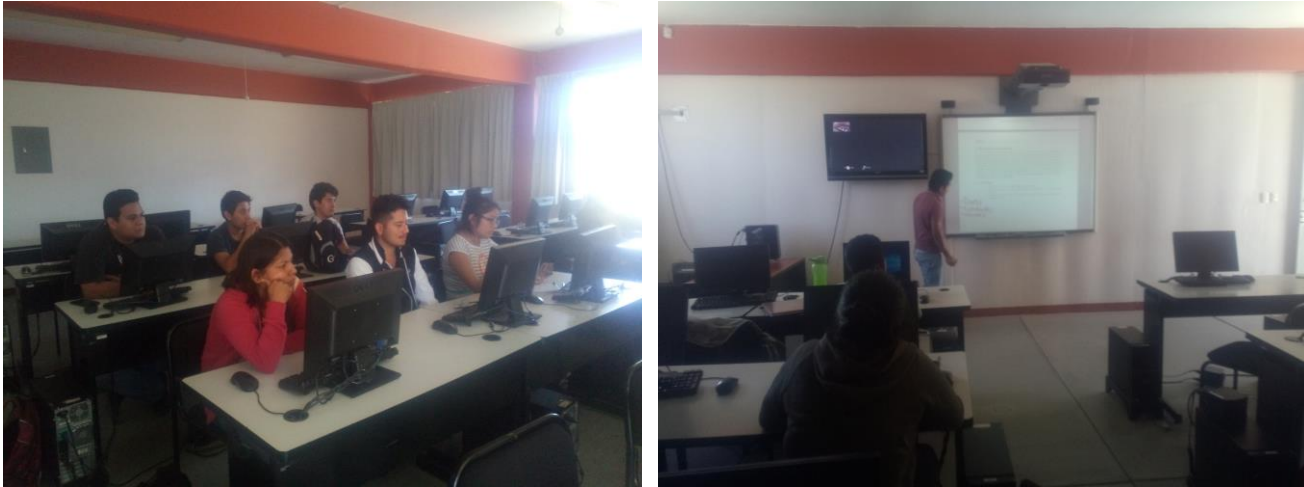


Figura 4.2. Estudiantes identificaron a los roles involucrados en el ejercicio

De manera similar, los estudiantes realizaron ejercicios en equipo en donde aplicaron las técnicas que utilizarían en Cadxela GSD. En este contexto, la Figura 4.3 muestra cómo los estudiantes realizaron un ejercicio en donde se les pidió aplicar la técnica de estimación por póker para valorar el tamaño de las historias de usuario para un problema en particular. Durante la ejecución del ejercicio, los estudiantes se mostraron entusiasmados y debatieron acerca de los diferentes puntos de vista y percepciones que tenían acerca del problema. Al final se obtuvo una lista de historias de usuario estimadas.



Figura 4.3. Estudiantes aprendiendo a aplicar la técnica de estimación por póker

Después, se abordó el tema de la gestión del conocimiento y su importancia en los entornos de GSD. En este caso se pidió a los estudiantes que investigaran sobre las estrategias de gestión del conocimiento que pudieran aplicarse en entornos globales y se debatieron sus hallazgos en clase.

Por último, durante los siguientes dos días del curso se impartieron los temas de colaboración y diversidad cultural en entornos de GSD. De igual modo, los estudiantes realizaron actividades de investigación sobre el impacto de ambos factores en un entorno de GSD. Finalmente, el último día del curso se resolvieron las dudas de los estudiantes acerca de los conceptos vistos en clase y se aplicó la pre-evaluación.

En resumen, el curso teórico que se impartió consistió no solamente de la transmisión de conceptos teóricos, sino también de la ejecución de ejercicios prácticos en donde los estudiantes aplicaban los conceptos vistos en clase. De esta manera se siguieron los lineamientos del marco de referencia que sugiere la capacitación de los estudiantes en el GSD siguiendo un enfoque constructivista. Por último, los estudiantes respondieron el instrumento de pre-evaluación de conocimientos sobre el GSD al terminar el curso (véase Figura 4.4).

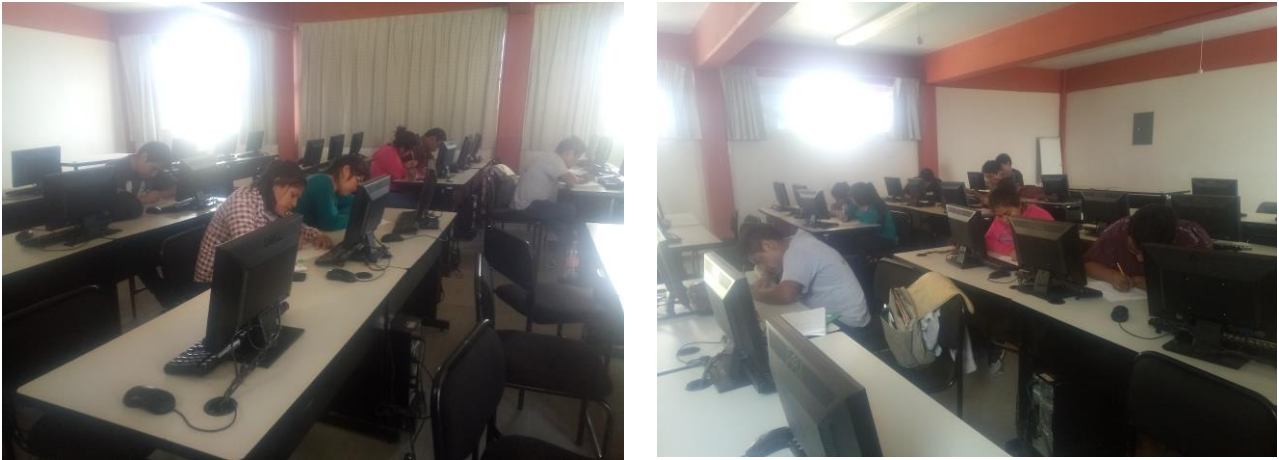


Figura 4.4. Estudiantes respondiendo el cuestionario de pre-evaluación

4.1.3.3. Ejecución de los proyectos de GSD

Una vez terminado el curso teórico, se organizaron los equipos de estudiantes que desarrollarían los proyectos de GSD a través de Cadxela GSD. Como se mencionó anteriormente, cada equipo estuvo conformado por cuatro estudiantes de las diferentes universidades participantes, por lo cual se formaron cinco equipos de trabajo. Además, a cada uno de éstos se les asignó un profesor responsable para que los apoyara durante el desarrollo de los proyectos. Posteriormente, cada profesor asignó el rol de “líder de proyecto” a un estudiante de su equipo y un proyecto de desarrollo de software de los ocho pre-cargados en Cadxela GSD (véase Tabla 13).

Tabla 13. Proyectos pre-cargados en Cadxela GSD.

Proyecto	Descripción
Sistema para el control de calificaciones escolares	El departamento de servicios escolares de la preparatoria “El Caudillo Veloz” incluye dentro de sus funciones el registro de estudiantes que cursarán estudios dentro de sus instalaciones y, por ende, realizar el proceso de registro de calificaciones de éstos durante su estancia en la preparatoria. Actualmente este proceso es realizado de manera manual por el personal del departamento, lo cual genera una ardua y tediosa labor administrativa tanto para los empleados del departamento como para los profesores. Actualmente la preparatoria almacena en archivos de Excel© la información de sus estudiantes y sus calificaciones. Sin embargo, el proceso para la captura de calificaciones es demasiado lento, ya que cada mes los profesores deben realizar evaluaciones a sus estudiantes y posteriormente registrarlas en un archivo. Cuando los profesores han terminado de capturar las calificaciones de sus estudiantes deben enviarlas por correo electrónico al departamento de servicios escolares para que se agregue esta información a los expedientes de los mismos. Por otro lado, si el estudiante desea saber la calificación que ha obtenido en cada una de sus asignaturas debe acudir a la oficina de sus profesores para preguntar acerca de ésta y, con base en esa información, obtener su promedio. Debido a lo anterior, la preparatoria ha decidido implementar un software que automatice este proceso permitiendo que los profesores registren directamente desde su oficina la información de la calificación de los estudiantes y, que a través de este medio se notifique a servicios escolares para que validen la información. Al mismo tiempo se pretende que se dé acceso a los

Proyecto	Descripción
	estudiantes con el objetivo de que puedan consultar sus calificaciones.
Sistema para el control de inventario	<p>La ferretería “Don Cheto” es una empresa que cuenta con más de 10 años de experiencia en el mercado ofreciendo a sus clientes productos de ferretería a un buen precio. Cada inicio de mes la empresa adquiere mercancía para ofrecer un catálogo moderno. Sin embargo, es difícil para los empleados saber qué productos están por terminarse en almacén o la cantidad exacta con la que se cuenta, ya que la empresa ha crecido y actualmente manejan una gran cantidad de productos. Debido a esto, la empresa requiere de un sistema computarizado donde se controle realmente la mercancía que entra y sale del almacén. Con dicho sistema se evitarían los problemas que se presentan a menudo, como por ejemplo que se agote la existencia de algún producto. También se evitaría que cualquier mercancía fuera robada y no se note su ausencia. Actualmente, la empresa cuenta con dos empleados en almacén y dos vendedores dentro de la tienda. En este sentido, los empleados del almacén deben monitorear constantemente la existencia de los productos. Cuando un producto esté por agotarse deben informar al dueño para que éste realice el pedido de la mercancía al proveedor. Sin embargo, en muchas ocasiones los empleados no se percatan de la baja disponibilidad de un producto o se les olvida avisar al dueño, lo cual ha ocasionado problemas cuando los vendedores ofrecen el producto a un cliente, pero no existe en almacén. Por lo tanto, la ferretería ha solicitado un software de control de inventario que permita controlar la mercancía de la ferretería y automatice el proceso de adquisición y baja de productos.</p>
Sistema para la gestión de préstamos de biblioteca	<p>Actualmente la biblioteca escolar de la preparatoria “Chicharito Hernández” ofrece el servicio de préstamos de libros a estudiantes por un periodo de, a lo más, 3 días. El proceso que se lleva a cabo para que un estudiante solicite un préstamo es el siguiente: el estudiante debe acudir directamente a la biblioteca y acercarse a alguno de los empleados para solicitar un préstamo, para ello, el empleado le preguntará acerca del libro que desea y éste proporcionará el título y autor del libro. Después, el empleado busca el libro y, si lo encuentra, solicita al estudiante su número de identificación escolar. Posteriormente, el empleado verifica en sus registros que el estudiante no cuente con adeudos (tiene un libro prestado actualmente, debe un libro por extravío o daño) en la biblioteca para poder otorgarle el préstamo. En caso de encontrar adeudos se le niega el préstamo. En caso contrario, el empleado llena un formulario con la información del estudiante y el libro que se le está prestando para guardarlo en los registros de la biblioteca. La biblioteca almacena en archivos la información de los préstamos que realiza y los adeudos de los estudiantes, por lo cual, el proceso para la gestión de préstamos es tedioso y ambiguo. Debido a esta forma de trabajo, la biblioteca ha enfrentado diversos problemas como: otorgar préstamos a estudiantes con adeudos por no tener un buen control sobre éstos, pérdida de libros por no llevar un correcto control sobre los adeudos, mal manejo de las fechas de vencimiento de los préstamos, entre otros. Debido a lo anterior, surge la necesidad de desarrollar un software que permita gestionar los préstamos de la biblioteca. El software que la biblioteca necesita deberá gestionar de manera eficaz a los usuarios de la biblioteca, los préstamos que solicitan los estudiantes, las fechas de vencimiento de los préstamos, y el control de los adeudos.</p>
Sistema para la gestión de un curso de inglés	<p>Como parte de las asignaturas formativas que se imparten en la preparatoria “Mi Escuelita” se tiene un curso de inglés que todos los estudiantes deben tomar. En este sentido, el profesor de esta asignatura ha solicitado un sistema que sirva de apoyo en la asignación y revisión de tareas y contenido de la materia. Este sistema deberá permitir que el profesor suba contenido audiovisual que podrá compartir con sus estudiantes. Es importante mencionar que en la preparatoria se cuenta con tres niveles de inglés: principiante, intermedio, y avanzado, y cada uno de los estudiantes se ubica en alguna de estas categorías. Por lo tanto, el material no es el mismo para todos los estudiantes, sino que se organiza de acuerdo con el nivel de éstos. Por otro lado, es deseo del profesor también el compartir ejercicios a los estudiantes y que éstos suban los ejercicios resueltos de igual manera a través del sistema. Finalmente, se debe permitir asignar una calificación a los ejercicios enviados por lo estudiantes. Este sistema permitirá a la escuela tener un mejor control sobre sus estudiantes, ya que actualmente se tiene a un número importante de éstos y solamente se tiene a dos profesores para impartir dicha</p>

Proyecto	Descripción
	materia.
Sistema para la planificación de reuniones	<p>La empresa “El Comodín” se dedica a ofrecer servicios de venta y fabricación de almohadas y tiene presencia en diferentes partes del mundo. Debido a esto, la empresa cuenta con socios y trabajadores ubicados en diferentes sitios. Recientemente la empresa se ha enfrentado a diversos problemas con relación a la organización de reuniones entre los socios y empleados. En cada reunión se toman decisiones acerca del futuro de la empresa o bien acerca del trabajo que realicen en cada una de las sucursales. En este sentido, los principales problemas que se presentan en la organización de reuniones son:</p> <ul style="list-style-type: none"> • Falta de disponibilidad de los participantes. • Cambios constantes en las restricciones de las reuniones (fecha, hora, lugar). • Lentitud en la organización de las reuniones. • Falta de mecanismos de realimentación para conocer la disponibilidad de los participantes. <p>Generalmente el proceso para organizar una reunión es el siguiente: el organizador de la reunión informa a los participantes que serán invitados sobre la necesidad de ésta y especifica un rango de fechas posibles para llevarla a cabo, entonces cada uno de los invitados responde con sus restricciones de disponibilidad (fecha, hora, y lugar) para participar en la reunión, con respecto a las fechas proporcionadas por el organizador. En la respuesta de los invitados se puede incluir la fecha, hora y lugar preferida por el invitado, aún si éstas no están dentro del rango proporcionado inicialmente por el organizador. Durante la organización de la reunión se pueden presentar conflictos de fechas, hora, y lugar de la reunión y es tarea del organizador contactar constantemente a los participantes para llegar a un acuerdo acerca de esta información. Algunas de las soluciones que se han tomado para solucionar los conflictos de fecha han sido extender el rango de fechas y volver a preguntar por la disponibilidad. Sin embargo, ya que este proceso se realiza actualmente de manera manual, el proceso es lento y tedioso. Por lo tanto, la empresa ha solicitado el desarrollo de un software que permita organizar las reuniones de manera efectiva. Para ello, se necesita que el software permita la planificación de las reuniones. La planificación debe incluir el invitar a los participantes para que establezcan sus restricciones de fecha, hora y lugar de la reunión. Este software deberá agilizar la organización de las reuniones y permitirá al organizador resolver conflictos de fecha, hora, y lugar al proveer un mecanismo de comunicación efectiva entre éste y cada uno de los participantes. Como resultado del proceso de organización se tendrá la planeación de una reunión con fecha, hora, y lugar que deberá ser aceptada y compartida con todos los participantes invitados a ésta.</p>
Sistema para la organización de un torneo de fútbol siete	<p>Como parte de las actividades deportivas que se organizan en la empresa de venta de equipo de cómputo “CompuLi”, se organiza un torneo de fútbol 7 entre los empleados de la empresa. El torneo se lleva a cabo con la participación de equipos organizados por los empleados con un máximo de 9 jugadores por equipo. Debido a lo anterior, la empresa deberá primeramente registrar a los equipos que participarán en el torneo y con base en esa información deberán organizar las jornadas de juego estableciendo los equipos que se enfrentarán, la hora y lugar en el que se llevará a cabo el partido. Así mismo, al final de cada partido se debe registrar la información de las estadísticas del juego: goles, jugadores que anotaron, tarjetas amarillas, y tarjetas rojas. Por último, es importante también que se lleve registro de las estadísticas del torneo, puntos obtenidos por cada equipo, goleadores del torneo, y por ende quién será el campeón. Para elegir al campeón se estableció una modalidad de final en donde se enfrentan los dos primeros lugares de la tabla general de posiciones. Por lo tanto, la empresa ha solicitado un software que le permita llevar a cabo la organización y control del torneo.</p>
Sistema para la venta de boletos de autobús	<p>La empresa de transporte de pasajeros terrestre “El Halcón” desea optimizar su proceso de venta y reservación de boletos con el objetivo de mejorar el servicio ofrecido a sus clientes, al mismo tiempo que agilizan y tienen mejor control de sus ventas. Para ello, han solicitado el desarrollo de un software de ventas, el cual sea capaz de llevar a cabo el proceso de venta y reserva de boletos. Actualmente el proceso del negocio de ventas de boletos es responsabilidad</p>

Proyecto	Descripción
	<p>del departamento de ventas y se lleva a cabo a través de las siguientes actividades:</p> <ul style="list-style-type: none"> • Registrar clientes. • Generar un boleto de viaje que muestre el número de asiento y el precio. • Reservar boletos a clientes. • Administrar las ventas. • Cancelar boletos reservados. <p>En este sentido, la venta de boletos se lleva a cabo de la siguiente manera: un cliente ingresa a la oficina de ventas y se acerca a la recepcionista, después pregunta por la disponibilidad y precio de los boletos al lugar a donde desea ir (la empresa ofrece los siguientes destinos: Oaxaca, Puebla, y Ciudad de México). Enseguida el recepcionista ofrece al cliente las opciones disponibles de viaje con los que cuenta, incluyendo el precio del boleto, los autobuses y asientos disponibles, y la hora de salida de cada uno de los autobuses. En este momento, el cliente puede retirarse sin comprar el boleto. En el caso de que decida comprar un boleto deberá proporcionar sus datos personales (nombre completo y lugar de salida), elegir un autobús, el asiento, y pagar. Finalmente, se le entregará su boleto y el recepcionista guardará el comprobante de la venta. Por otro lado, la reservación de boletos se puede llevar a cabo directamente en la oficina de ventas o por medio de una llamada telefónica. En este caso, el proceso es similar a la venta de boletos: El cliente solicita información del viaje que desea realizar y el recepcionista le ofrecerá detalles acerca de los autobuses, asientos, y fechas disponibles para su viaje. Después el cliente elige el autobús, asiento, y fecha. Finalmente, el recepcionista registra en su libreta de reservaciones los datos personales y la información de la reservación. Por otro lado, el cliente puede cancelar su reservación por medio de una llamada telefónica. Para hacer valida la reservación el cliente debe presentarse en la oficina de ventas 30 minutos antes de la salida del autobús para realizar el pago del boleto. Si no realiza el pago en ese periodo de tiempo, el recepcionista puede vender el boleto reservado. Esta forma de venta y recepción ha funcionado de manera correcta en la empresa desde su fundación; sin embargo, la empresa ha crecido y el proceso se ha vuelto obsoleto. Por esta razón, la empresa ha decidido implementar un software que automatice el proceso resumido anteriormente.</p>
Sistema para la venta y renta de películas	<p>El videoclub “Video-Lab” es una empresa dedicada a la venta y renta de películas en formato DVD. Actualmente la empresa ofrece sus servicios de venta y renta a sus clientes a través de pedidos en tienda. Es decir, los clientes acuden a la tienda físicamente y solicitan a los empleados la compra o renta de alguna película. De manera general, el proceso para la venta y renta de películas es el siguiente: un cliente ingresa a la tienda y solicita una película por medio del título de ésta. El empleado busca en su computadora la película y si es encontrada ofrece al cliente la opción de venta o renta. En el caso de la venta, el cliente debe proporcionar su información personal (nombre completo) y pagar por el precio para que el empleado registre la venta en una hoja de Excel©. Para el caso de las rentas, el cliente debe registrarse como cliente de la tienda (en caso de no serlo aún) a lo cual el empleado le otorgará un identificador de cliente y registrará la información de la película rentada asociándola con el cliente. Las películas se pueden rentar a los clientes solamente si éstos no han rentado más de dos películas y no tienen adeudos con el video club. En este sentido, se considera un adeudo cuando el cliente supera los dos días del préstamo y no entrega la película, así mismo, es un adeudo si daña el DVD o lo extravía. El proceso que lleva el video club actualmente para la venta y renta de películas es lento con relación a la cantidad de clientes que debe atender por día. Debido a esto la empresa ha decidido implementar un software que ofrezca los servicios de venta y renta de películas a través de internet. De esta manera, los clientes podrán registrarse en el sistema y realizar la compra o renta de sus películas. Así, cuando un cliente realice una compra o renta se le enviará la película a su domicilio. Otro objetivo que se pretende alcanzar con la implementación del software es tener mejor control de los adeudos de los clientes, notificándoles cuando el préstamo que han solicitado haya vencido y permitiendo a los empleados registrar adeudos por daño o extravío.</p>

En este sentido, los proyectos que contiene el repositorio de la plataforma fueron diseñados con el objetivo de describir sistemas pequeños que pudieran ser desarrollados en un periodo no mayor a tres semanas, con la finalidad de que los estudiantes fueran capaces de terminarlos en tiempo y forma. Dicha duración fue calculada utilizando el método de estimación de Puntos de Caso de Uso (UCP, por sus siglas en inglés) propuesto por Karner en 1993. Dicho método utiliza a los casos de uso como variables de entrada. Los casos de uso son una forma popular de representar a los requisitos funcionales de un sistema y su principal objetivo es representar la interacción entre el usuario final (llamado actor) y el sistema, en términos de transacciones, usando lenguaje natural. El método de UCP inicia con la evaluación de la complejidad de los actores y los casos de uso, para después ajustar los valores obtenidos con los factores de caracterización del entorno de desarrollo y la complejidad técnica del sistema a desarrollar (Ochodek, Nawrocki, y Kwarciak, 2011).

A continuación, a modo de ejemplo se muestra la estimación realizada para el proyecto “Sistema para la organización de un torneo de fútbol siete” implementado el método de UCP. Por lo tanto, los casos de uso mostrados en la Tabla 14 fueron obtenidos por el tesista para determinar una estimación aproximada de la duración de dicho proyecto.

Tabla 14. Casos de uso para el sistema de organización de un torneo de fútbol siete.

#	Definición del caso de uso	
1	Nombre del caso de uso:	Registrar torneo
	Actor(es):	Administrador
	Escenario principal	Escenario secundario
	1. El administrador selecciona la opción registrar torneo. 2. El sistema muestra un formulario para el registro del torneo. 3. El administrador ingresa la información del torneo y envía el formulario. 4. El sistema informa que el torneo fue registrado.	3.1 El administrador no agregó toda la información del torneo. 3.2 El sistema muestra un mensaje de error.
2	Nombre del caso de uso:	Registrar equipo
	Actor(es):	Administrador
	Escenario principal	Escenario secundario
	1. El administrador selecciona la opción registrar equipo. 2. El sistema muestra un formulario para el registro del equipo. 3. El administrador ingresa la información del equipo y envía el formulario. 4. El sistema informa que el equipo ha sido registrado.	3.1 El administrador no agregó toda la información del equipo. 3.2 El sistema muestra un mensaje de error.
3	Nombre del caso de uso:	Registrar partido
	Actor(es):	Administrador
	Escenario principal	Escenario secundario
	1. El administrador selecciona la opción registrar partido. 2. El sistema muestra un formulario para el registro del partido. 3. El administrador ingresa la información del partido y envía el formulario.	3.1 El administrador no agregó toda la información del partido. 3.2 El sistema muestra un mensaje de error.

	4. El sistema informa que el partido ha sido registrado.	
4	Nombre del caso de uso:	Registrar estadísticas de torneo
	Actor(es):	Administrador
	Escenario principal	Escenario secundario
	1. El administrador selecciona la opción registrar estadísticas de torneo. 2. El sistema muestra un formulario para el registro de las estadísticas. 3. El administrador ingresa la información de las estadísticas y envía el formulario. 4. El sistema informa que los datos han sido registrados.	3.1 El administrador no agregó toda la información del torneo. 3.2 El sistema muestra un mensaje de error.
5	Nombre del caso de uso:	Registrar estadísticas de partido
	Actor(es):	Administrador
	Escenario principal	Escenario secundario
	1. El administrador selecciona la opción registrar estadísticas de partido. 2. El sistema muestra un formulario para el registro de las estadísticas. 3. El administrador ingresa la información de las estadísticas y envía el formulario. 4. El sistema informa que los datos han sido registrados.	3.1 El administrador no agrego toda la información del torneo. 3.2 El sistema muestra un mensaje de error.
6	Nombre del caso de uso:	Ver calendario del torneo
	Actor(es):	Administrador, Arbitro, Jugador
	Escenario principal	Escenario secundario
	1. El usuario selecciona la opción ver calendario del torneo. 2. El sistema muestra el calendario de partidos programados.	

El primer paso del método UCP es establecer un nivel de complejidad para cada uno de los actores que participan en el sistema, para ello el método proporciona las clases de complejidad que se muestran en la Tabla 15. De esta manera, en el sistema participan 3 actores: administrador, árbitro, y jugador, todos con un nivel de complejidad “complejo” por lo que el Peso de los Actores sin Ajustar (UAW, por sus siglas en inglés) es igual a 9, como se puede apreciar en la Tabla 16.

Tabla 15. Clases de complejidad de actores en el método de UCP.

Complejidad	Definición	Peso
Simple	Un sistema que se comunica con otros sistemas a través de un API.	1
Medio	Un actor del sistema que se comunica a través de un protocolo de comunicación (e.g., HTTP, FTP), o a una persona que interactúa a través de una consola de terminal.	2
Complejo	Una persona que utiliza una interfaz gráfica de usuario (GUI) para comunicarse con un sistema.	3

Tabla 16. UAW para el sistema de organización de un torneo de fútbol siete.

Actor	Tipo	Peso
Administrador	Complejo	3
Arbitro	Complejo	3
Jugador	Complejo	3
Total		9

El segundo paso del método consiste en evaluar el nivel de complejidad de los casos de uso, la cual dependerá del número total de transacciones identificadas para cada caso de uso. Una transacción es un conjunto de actividades en escenarios de casos de uso, que se realizan completamente o no se realizan en absoluto. La Tabla 17 muestra los niveles de complejidad de los casos de uso definidos en el método, así como el peso asociado a cada uno de éstos. Por otro lado, la Tabla 18 muestra la complejidad asociada a cada caso de uso identificado para el sistema, así como los Pesos de Casos de Uso sin Ajustar (UUCW, por sus siglas en inglés) obtenidos.

Tabla 17. Clases de complejidad de casos de uso en el método de UCP.

Complejidad	Definición	Peso
Simple	Contiene tres o menos transacciones	5
Medio	Contiene entre cuatro y siete transacciones	10
Complejo	Contiene más de siete transacciones	15

Tabla 18. UUCW para el sistema de organización de un torneo de fútbol siete.

Caso de uso	Tipo	Peso
Registrar torneo	Simple	5
Registrar equipo	Simple	5
Registrar partido	Simple	5
Registrar estadísticas de partido	Simple	5
Registrar estadísticas de torneo	Simple	5
Ver calendario del torneo	Simple	5
Total		30

El método de UCP incluye 13 factores de ajuste relacionados con la complejidad técnica del sistema a desarrollar, y 8 más relacionados con el entorno en el cual será desarrollado. El método define dos variables de ajuste: el Factor de Complejidad Técnica (TCF, por sus siglas en inglés) y el Factor de Complejidad del Entorno (EF, por sus siglas en inglés). La influencia de los TCF se evalúa asignando un valor de 0 a 5 a cada uno de los factores de complejidad técnica y multiplicándolo por un peso específico definido para cada uno de éstos (véase la Tabla 19), para finalmente aplicar la ecuación (1) y obtener así un TCF de 0.71 para este ejemplo.

Tabla 19. TCF para el sistema de organización de un torneo de fútbol siete.

Factor	Descripción	Peso	Impacto percibido	Factor calculado
T1	Sistema distribuido.	2	0	0
T2	Objetivos de performance o tiempo de respuesta.	1	1	1
T3	Eficiencia del usuario final.	1	2	2
T4	Procesamiento interno complejo.	1	0	0
T5	El código debe ser reutilizable.	1	1	1
T6	Facilidad de instalación.	0.5	0	0
T7	Facilidad de uso.	0.5	4	2
T8	Portabilidad.	2	1	2
T9	Facilidad de cambio.	1	2	2
T10	Concurrencia.	1	0	0
T11	Incluye objetivos especiales de seguridad.	1	0	0
T12	Provee acceso directo a terceras partes.	1	0	0
T13	Se requiere facilidades especiales de entrenamiento a usuario.	1	1	1
Total				11

$$TCF = 0.6 + (0.01 \times \sum_{i=1}^{13} TF_weight_i \times value_i) \quad (1)$$

En donde:

- TF_weight_i es el peso de factor de complejidad técnica en la i-ésima posición.
- $Value_i$ es el grado de influencia del factor de complejidad técnica en la i-ésima posición.

La influencia de los factores del entorno es evaluada de manera similar. A cada factor se le asigna un valor de 0 a 5 con base en su influencia percibida y se multiplica este valor por su peso. La Tabla 20 muestra los valores asignados a cada factor.

Tabla 20. EF para el sistema de organización de un torneo de fútbol siete.

Factor	Descripción	Peso	Impacto percibido	Factor calculado
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4	6
E2	Experiencia en la aplicación.	0.5	4	2
E3	Experiencia en orientación a objetos.	1	3	3
E4	Capacidad del analista líder.	0.5	4	2

Factor	Descripción	Peso	Impacto percibido	Factor calculado
E5	Motivación.	1	4	4
E6	Estabilidad de los requerimientos	2	4	8
E7	Personal de tiempo parcial	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	1	-1
Total				21

El EF se obtiene de aplicar la ecuación (2), para obtener un valor de 0.77.

$$EF = 1.4 + (-0.03 \times \sum_{i=1}^8 EF_weight_i \times value_i) \quad (2)$$

En donde:

- EF_weight_i es el peso de factor del entorno en la i -ésima posición.
- $Value_i$ es el grado de influencia del factor del entorno en la i -ésima posición.

Con esta información es posible obtener, primeramente, 39 Puntos de Casos de Uso sin Ajustar (UUCP, por sus siglas en inglés) al aplicar la ecuación (3). Posteriormente, al aplicar la ecuación (4) se obtienen 21 UCP para este proyecto.

$$UUCP = UAW + UUCW \quad (3)$$

$$UCP = UUCP \times TCF \times EF \quad (4)$$

Finalmente, para obtener una estimación de esfuerzo en horas-hombre, se debe multiplicar el UCP por un factor de productividad (PF, por sus siglas en inglés). El valor predeterminado definido por el método para el PF es de 20, por lo cual al utilizar este valor en la estimación se obtiene un total de 426 horas para desarrollar dicho proyecto. En este sentido, se sugiere que para cada proyecto los estudiantes dediquen un total de 35 horas por semana a la codificación, por lo cual, se tiene un total de 140 horas de trabajo por semana, de ahí que la duración estimada del proyecto “Sistema para la organización de un torneo de fútbol siete” haya sido poco más de 3 semanas. La aplicación del método UCP se realizó de la misma manera para el resto de los proyectos y al final se obtuvo un promedio de duración que fue manejado durante el caso de estudio.

La Figura 4.5 muestra el ejemplo de un proyecto asignado a uno de los equipos a través de Cadxela GSD. En el momento en que el profesor asigna un proyecto a su equipo, Cadxela GSD notifica automáticamente al líder de proyecto a través de un correo electrónico y un aviso en la sesión de la plataforma. A continuación, el equipo realiza la planificación del proyecto a través de Cadxela GSD mediante la aplicación de las técnicas ágiles vistas en el curso teórico, para después desarrollar el proyecto que se debe entregar al profesor responsable. Cabe aclarar que los profesores monitorean constantemente el trabajo de los estudiantes a través de Cadxela GSD, mientras que éstos planifican su trabajo, se comunican, y colaboran como equipo.

Durante la ejecución del caso de estudio se llevaron a cabo tres rondas de proyectos, lo cual permitió desarrollar en total 15 proyectos de GSD. Para cada ronda se organizaron diferentes

equipos de estudiantes y se asignó un nuevo profesor a cada uno de éstos, con el fin de que cada estudiante desempeñara un rol distinto para cada proyecto y trabajara con distintos compañeros. En adelante se describirán los resultados obtenidos para cada ronda de proyectos.

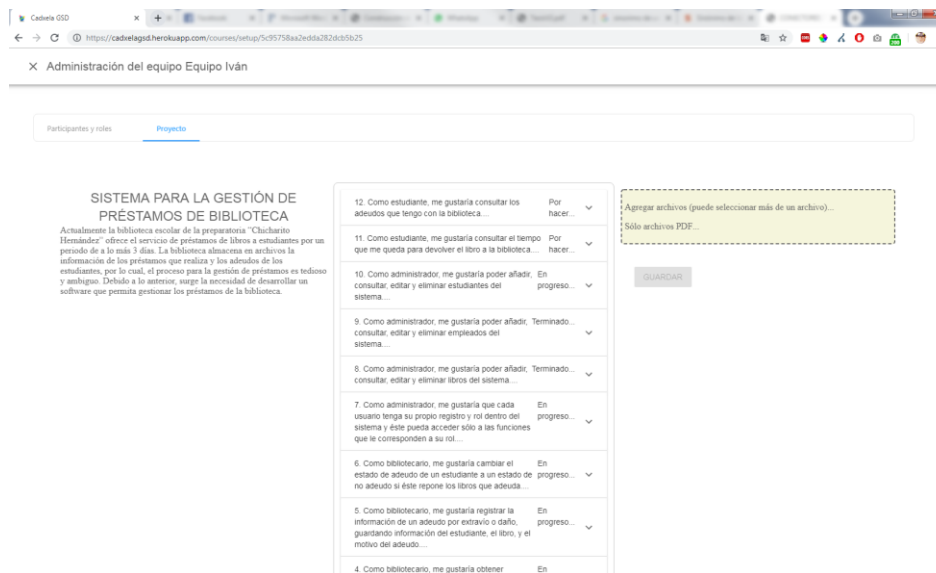


Figura 4.5. Proyecto asignado a un equipo en Cadxela GSD

4.1.3.3.1. Primera ronda de proyectos

Durante la primera asignación de proyectos, los equipos enfrentaron diversos problemas relacionados con las distancias geográfica y temporal, presentes en un entorno de GSD, para organizar su trabajo, las cuales se pueden resumir de la siguiente manera:

- Problemas para comunicarse entre los miembros del equipo: Al inicio del proyecto ni los estudiantes ni los profesores estaban relacionados con Cadxela GSD, ni habían trabajado en un entorno de GSD, lo cual provocó que no supieran como iniciar la comunicación con sus compañeros de trabajo. Así mismo, los estudiantes no comprendían sus funciones dentro del equipo. Concretamente, el líder del proyecto no realizaba sus actividades dentro de Cadxela GSD, por lo cual sus compañeros no trabajaban, ya que no sabían en qué estado del proyecto se encontraban.
- Problemas para establecer un horario en el cual todos pudieran trabajar: Otro problema al que se enfrentaron los estudiantes al inicio del proyecto fue el establecer un horario para llevar a cabo sus actividades. Debido a que todos los participantes llevaban a cabo actividades académicas propias de su institución, el trabajar en un horario laboral (e.g., de 9:00 a 19:00) resultaba complicado. Por lo tanto, los estudiantes tuvieron que reunirse por las noches; sin embargo, no todos podían llegar a tiempo para las reuniones y tenían que ponerse al corriente a través de los mensajes guardados en la plataforma.
- Problemas de coordinación: Los estudiantes tuvieron dificultades para organizar y coordinar las diferentes actividades que deberían realizar en Cadxela GSD. Lo anterior provocó que las actividades de planificación tardaran más de lo necesario en realizarse y que el tiempo para el desarrollo del software fuera de, a lo más, de una sola iteración de siete días.

- Problemas técnicos: Los estudiantes enfrentaron diversos problemas con relación a los aspectos técnicos del desarrollo de software, particularmente con los sistemas de control de versiones, y las técnicas ágiles implementadas en Cadxela GSD, ya que no conocían ni dominaban estas herramientas y técnicas.

Los problemas mencionados anteriormente provocaron que, para esta primera ronda de proyectos, se obtuvieran malos resultados con relación al desempeño de los equipos. En este sentido, las siguientes figuras muestran los resultados obtenidos por cada uno de los equipos durante esta ronda de proyectos.

Las gráficas muestran que en general ninguno de los equipos cumplió con las estimaciones realizadas (representadas por la línea color gris) y su desempeño en el desarrollo del proyecto fue demasiado bajo (representado por la línea azul). Posiblemente tal desempeño se deba a los problemas citados anteriormente. Otro aspecto que podría explicar el mal desempeño de los equipos es que durante la duración total del proyecto se reportaron pocas reuniones entre los equipos, concretamente para el caso del Equipo A solamente se llevaron a cabo siete reuniones, por lo cual es lógico pensar que no hubo una buena comunicación ni colaboración entre los integrantes del equipo.

Es importante considerar también que en el caso del Equipo E los estudiantes entregaron gran parte de las funcionalidades del producto, por lo tanto, es posible que, aunque los estudiantes estaban trabajando, no reportaban su avance a través de Cadxela GSD. En este contexto, la plataforma proporciona un tablero de tareas en donde cada integrante del equipo debe actualizar el estado de su trabajo con el objetivo de que el líder de proyecto pueda monitorear su participación. Así mismo, este equipo reportó un mayor número de interacciones con un total de once reuniones realizadas durante la duración del proyecto.

En definitiva, ninguno de los equipos obtuvo buenos resultados, ya que no pudieron entregar el proyecto en tiempo y forma con base en sus estimaciones. También es importante destacar que la comunicación y colaboración de los equipos fue deficiente, en gran medida porque estaban relacionándose tanto con el enfoque de GSD como con las diferentes herramientas que utilizaron para trabajar (e.g., sistemas de control de versiones, Cadxela GSD, lenguajes y técnicas de programación, técnicas ágiles, etc.).

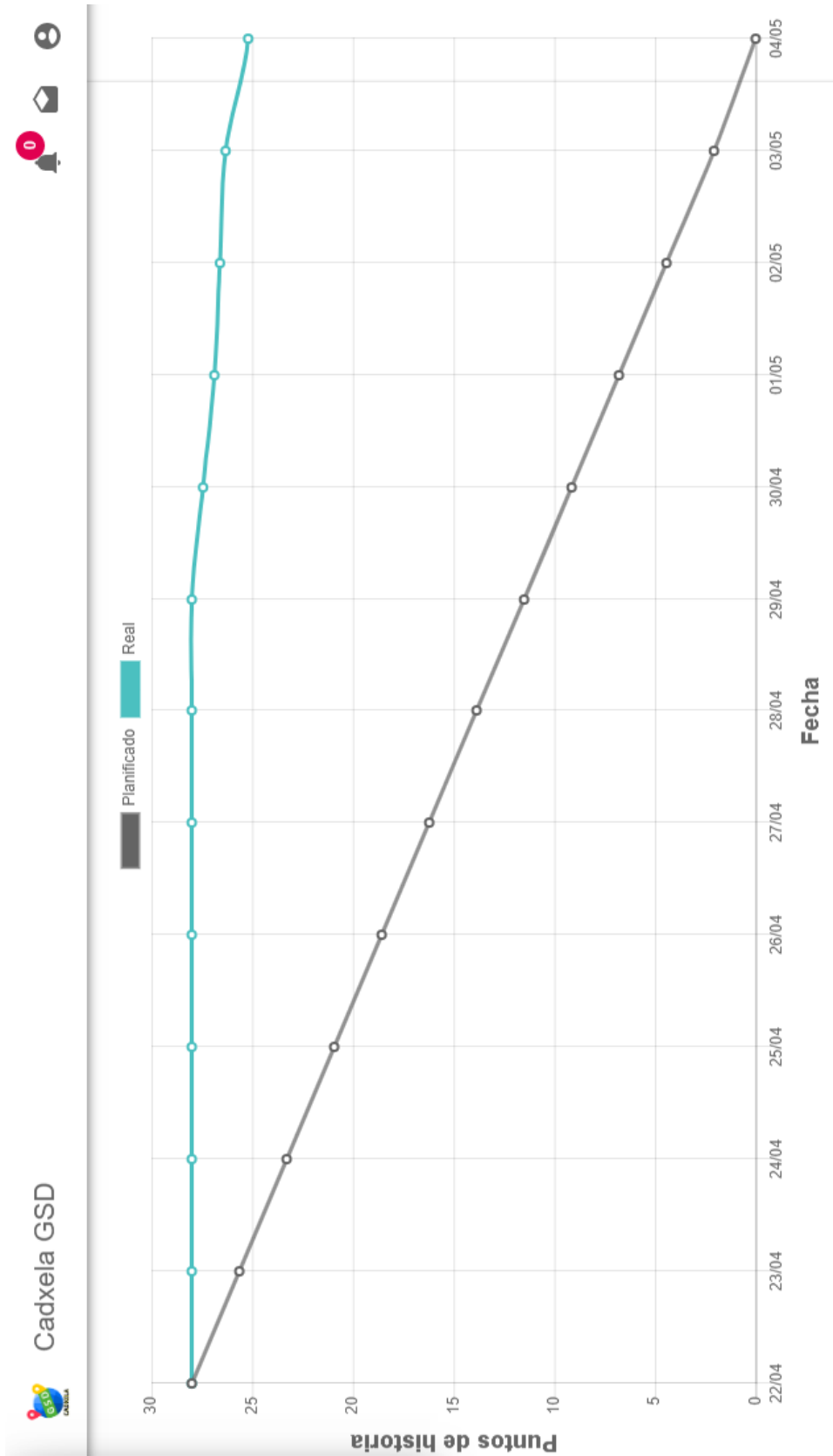


Figura 4.6 Desempeño del equipo A en la primera ronda de proyectos

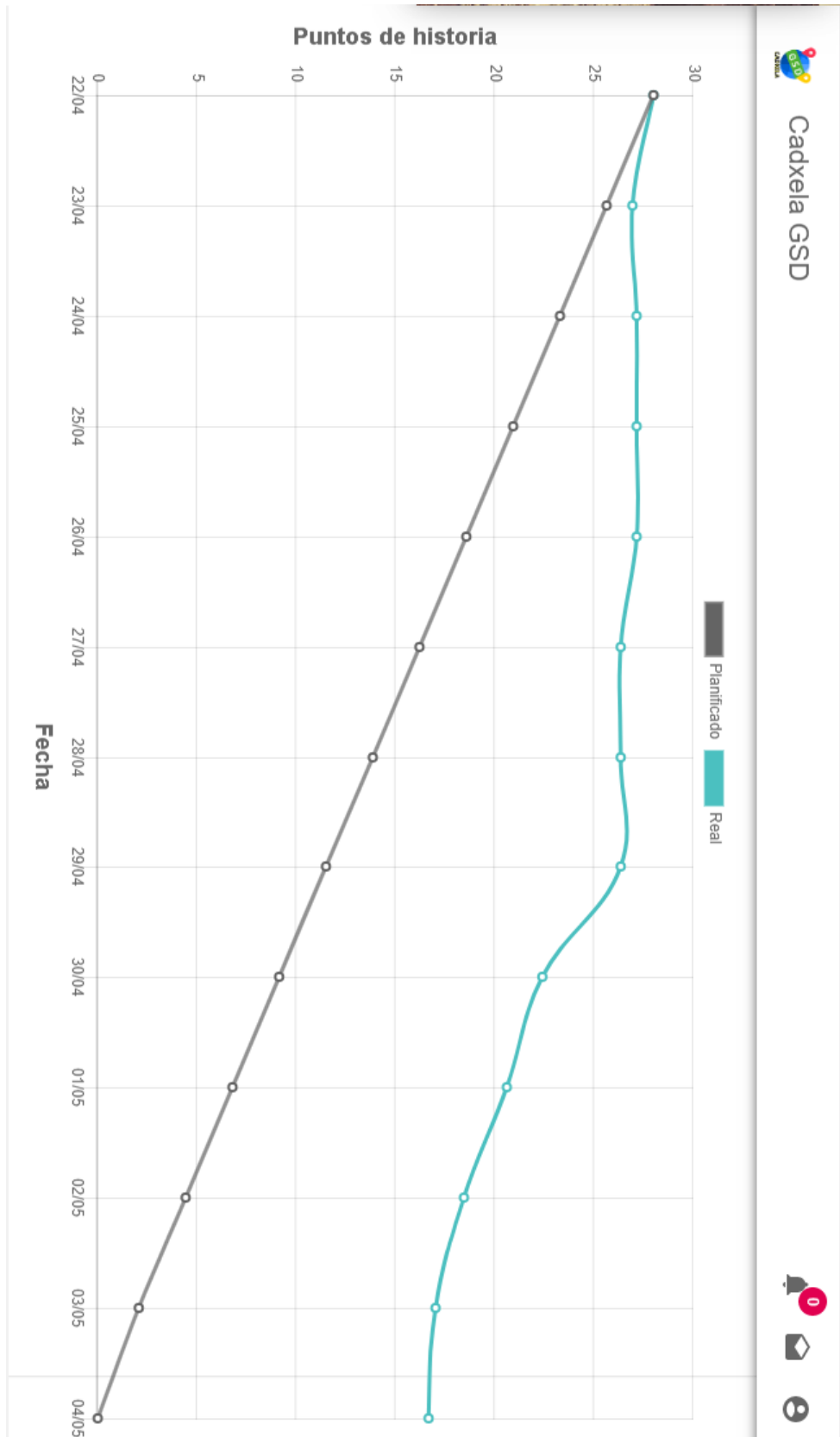


Figura 4.7 Desempeño del equipo B en la primera ronda de proyectos.

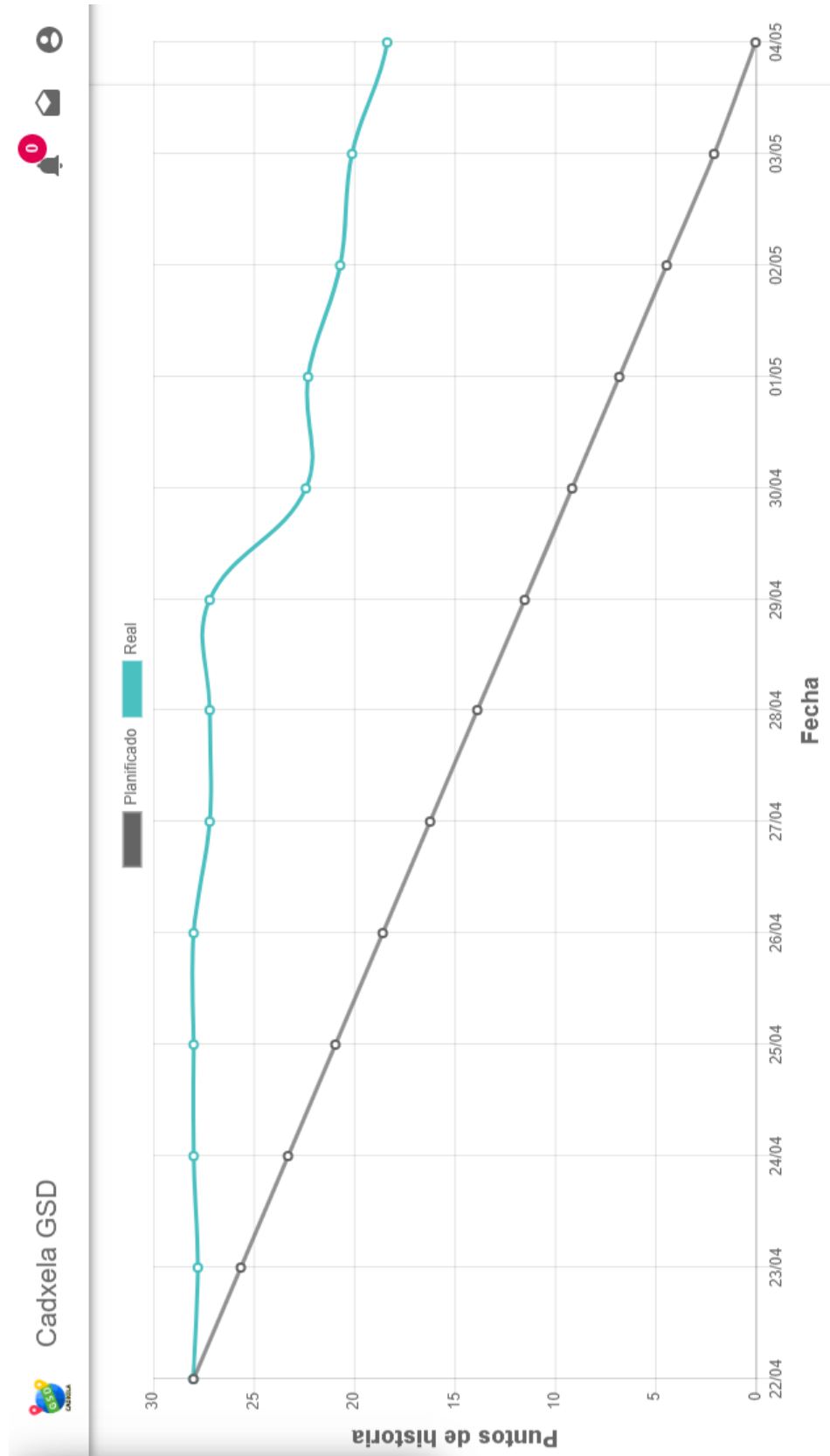


Figura 4.8 Desempeño del equipo C en la primera ronda de proyectos.

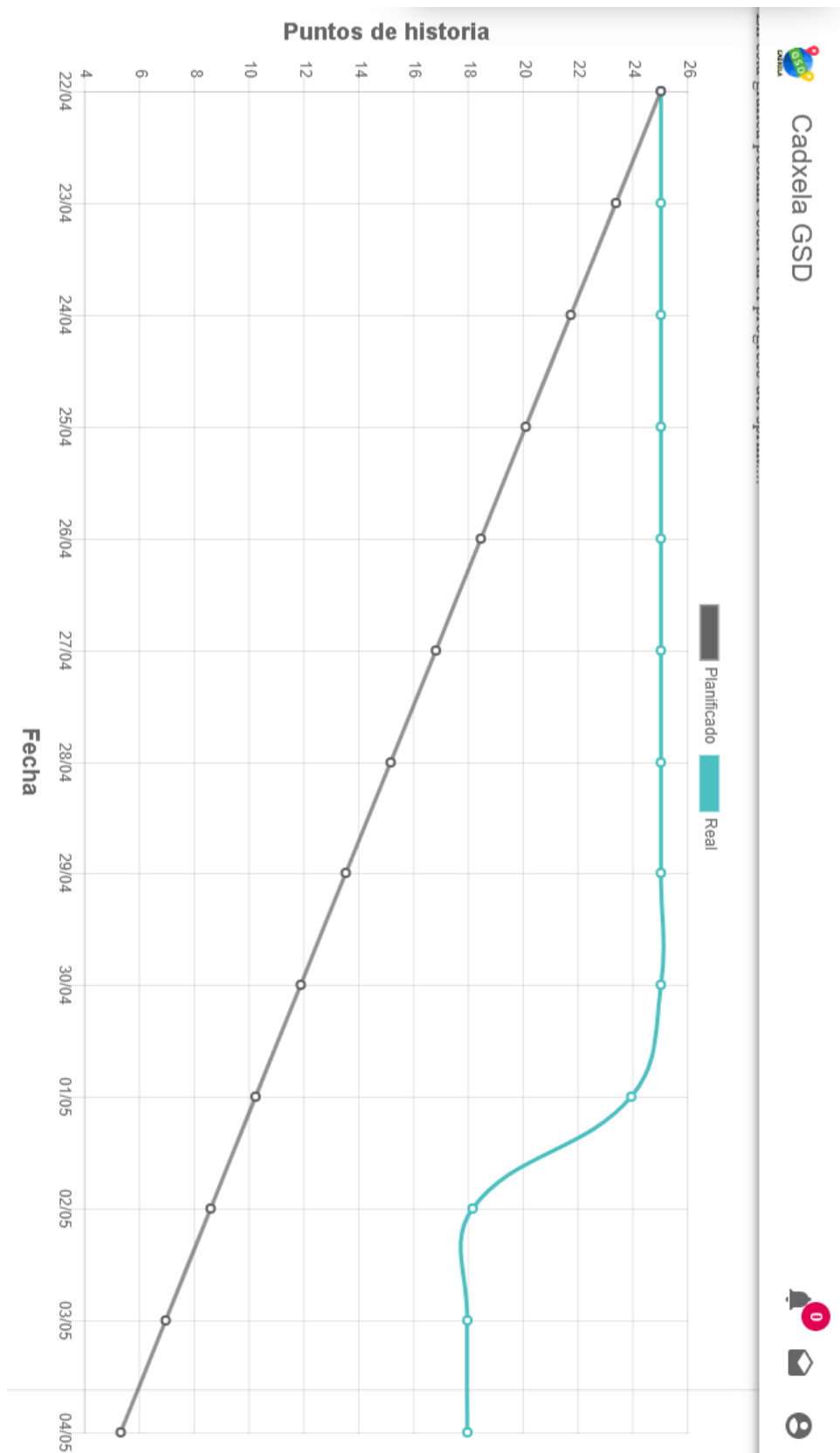


Figura 4.9 Desempeño del equipo D en la primera ronda de proyectos.

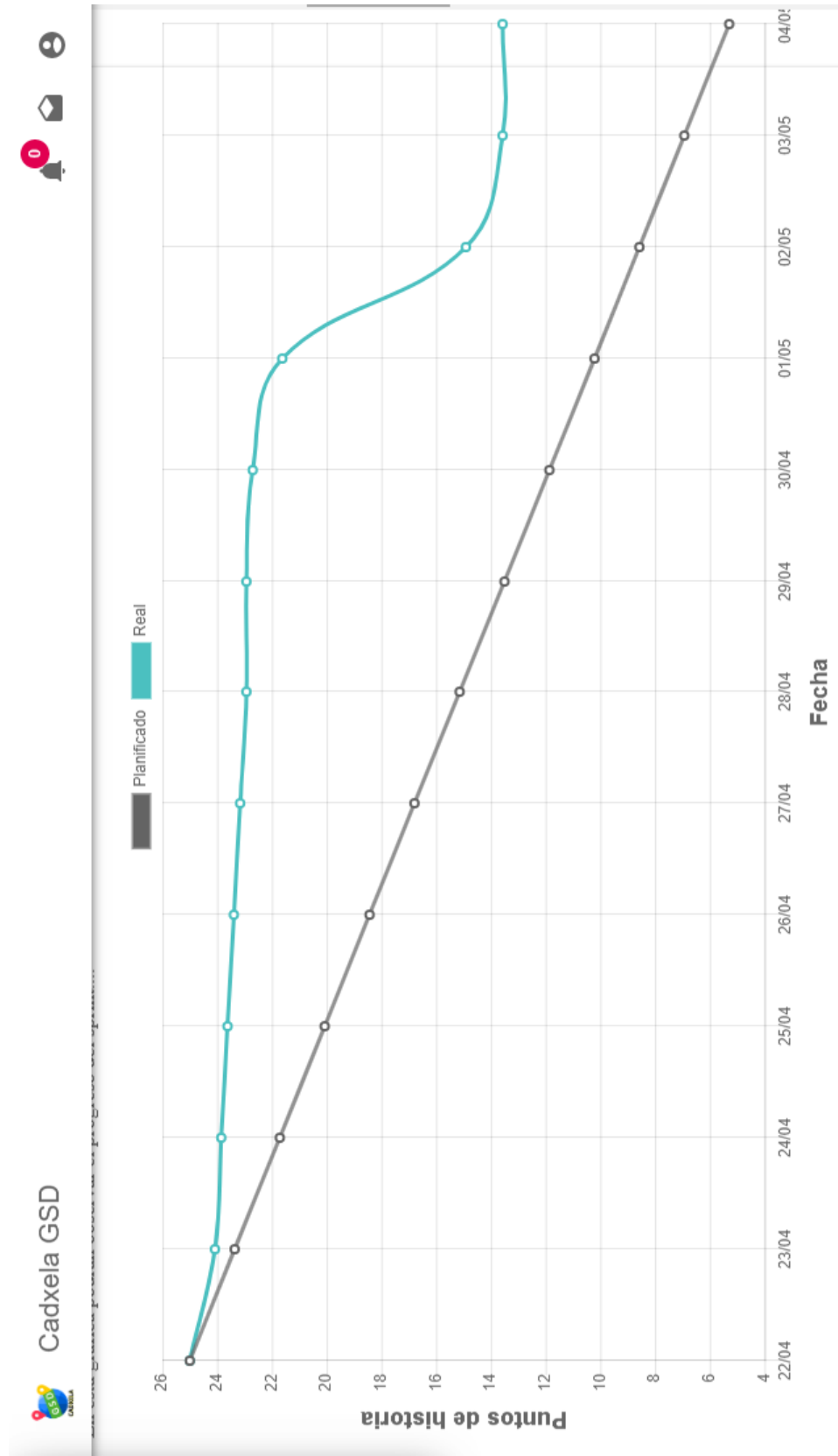


Figura 4.10 Desempeño del equipo E en la primera ronda de proyectos.

4.1.3.3.2. Segunda ronda de proyectos

Para la segunda ronda de proyectos se reorganizaron los equipos y se asignó un nuevo profesor a cada uno de éstos. Nuevamente se asignaron roles y se distribuyeron nuevos proyectos. En cuanto al desempeño de los equipos, se puede afirmar que éste mejoró considerablemente respecto a los equipos de la primera ronda. Para este momento los equipos ya estaban relacionados un poco más con el enfoque de GSD y tenían más experiencia en la aplicación de las técnicas ágiles presentes en Cadxela GSD. Además, con el primer proyecto los estudiantes aprendieron a utilizar los sistemas de control de versiones y comprendieron la importancia de establecer una buena comunicación y colaboración con los miembros de sus equipos.

Con respecto a la comunicación y el trabajo en equipo, es posible afirmar que éstos mejoraron notablemente debido a las nuevas acciones realizadas por los estudiantes. En este sentido, la organización del trabajo y el establecimiento de horarios para trabajar requirió de un menor esfuerzo que en la ronda anterior, ya que los equipos realizaron su estimaciones y planificación en un tiempo menor, teniendo así iteraciones más largas para la construcción de los sistemas. Ahora bien, los principales problemas enfrentados por los estudiantes en esta segunda ronda fueron los siguientes:

- Falta de trabajo en equipo: Durante la duración del proyecto muchos estudiantes estuvieron ausentes en las reuniones del equipo, en general fue difícil que todos los integrantes de un equipo estuvieran presentes en una misma reunión. Lo anterior provocó que el trabajo en equipo fuera deficiente, sobre todo al inicio del proyecto cuando se realizaba su planificación. Aunado a lo anterior, durante la mayoría de las reuniones la toma de decisiones recaía en el líder de proyecto y los otros miembros del equipo limitaban su participación a brindar pocas opiniones.
- Problemas técnicos con las herramientas de control de versiones: Los estudiantes presentaron diversos problemas con los sistemas de control de versiones, aunque ya habían trabajado con éstas en la ronda anterior, puesto que seguían sin entender correctamente cómo funcionan estos sistemas. Por lo tanto, tuvieron que tomar un tiempo extra para aprender un poco más.

Si bien los resultados obtenidos en esta ronda mejoraron, todavía se presentaron diversos problemas y el desempeño de los equipos no fue del todo óptimo. Para ilustrar mejor, las siguientes figuras muestran los resultados obtenidos por los equipos en la segunda ronda.

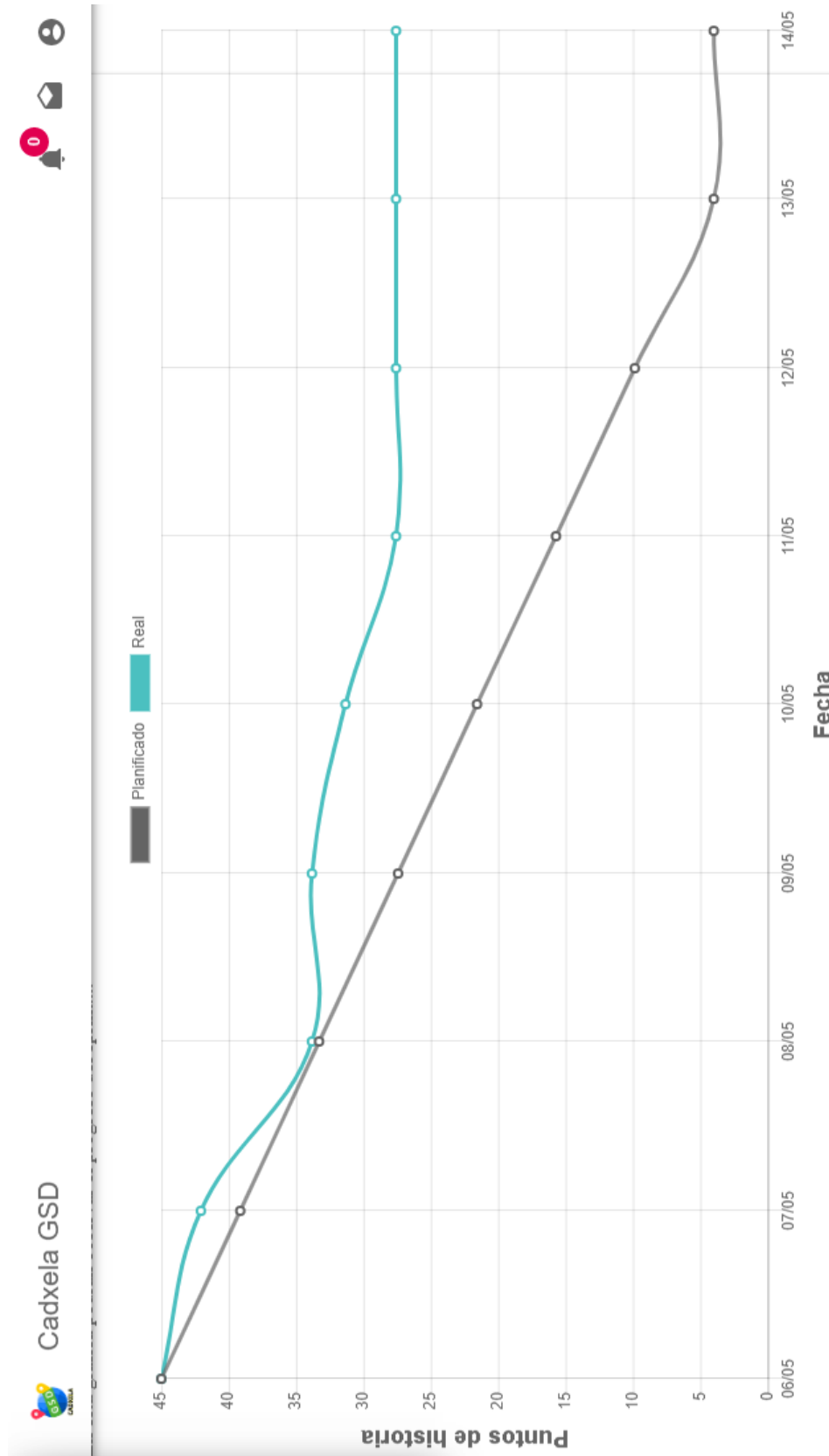


Figura 4.11 Desempeño del equipo A en la segunda ronda de proyectos.

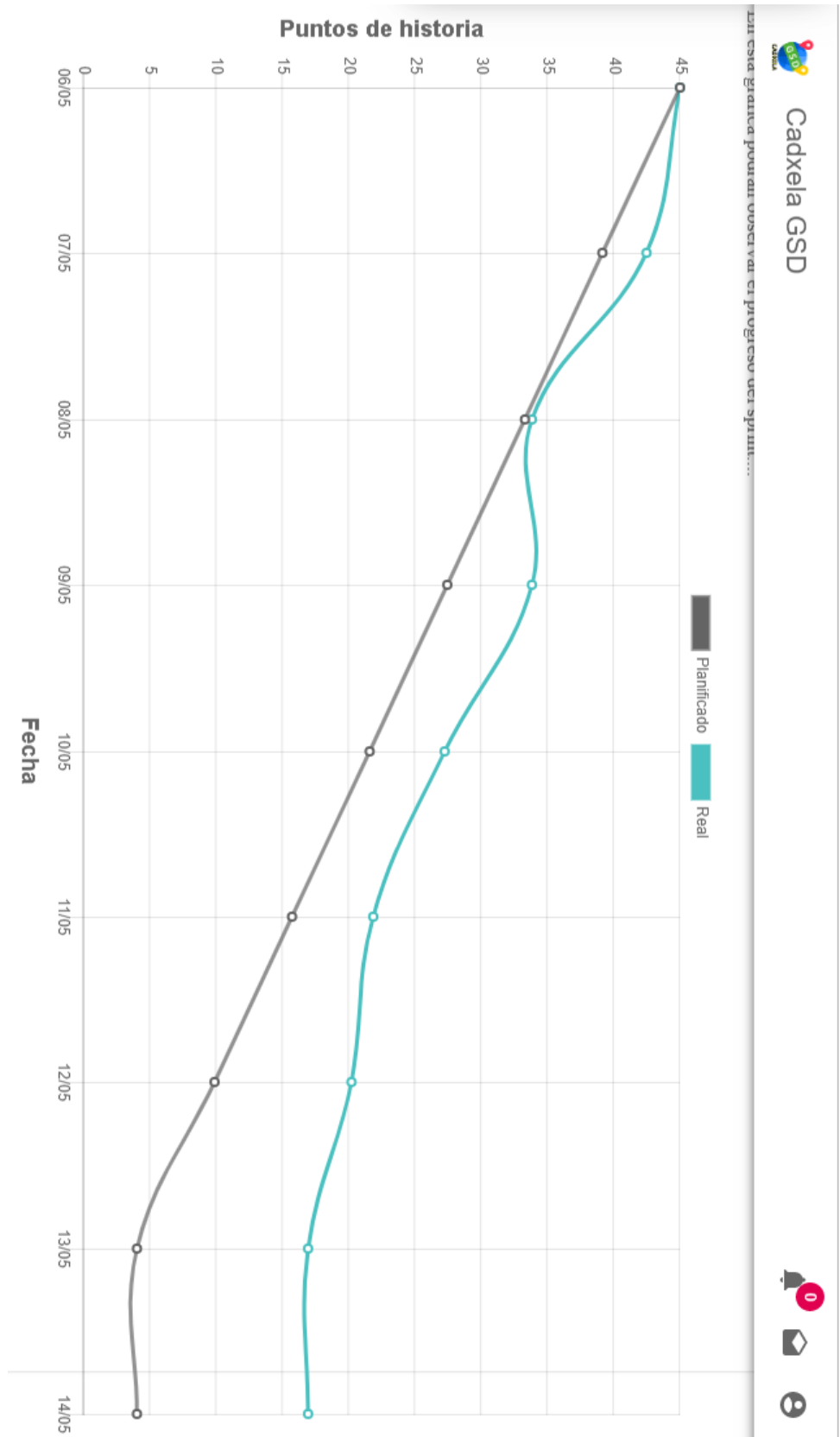


Figura 4.12 Desempeño del equipo B en la segunda ronda de proyectos.

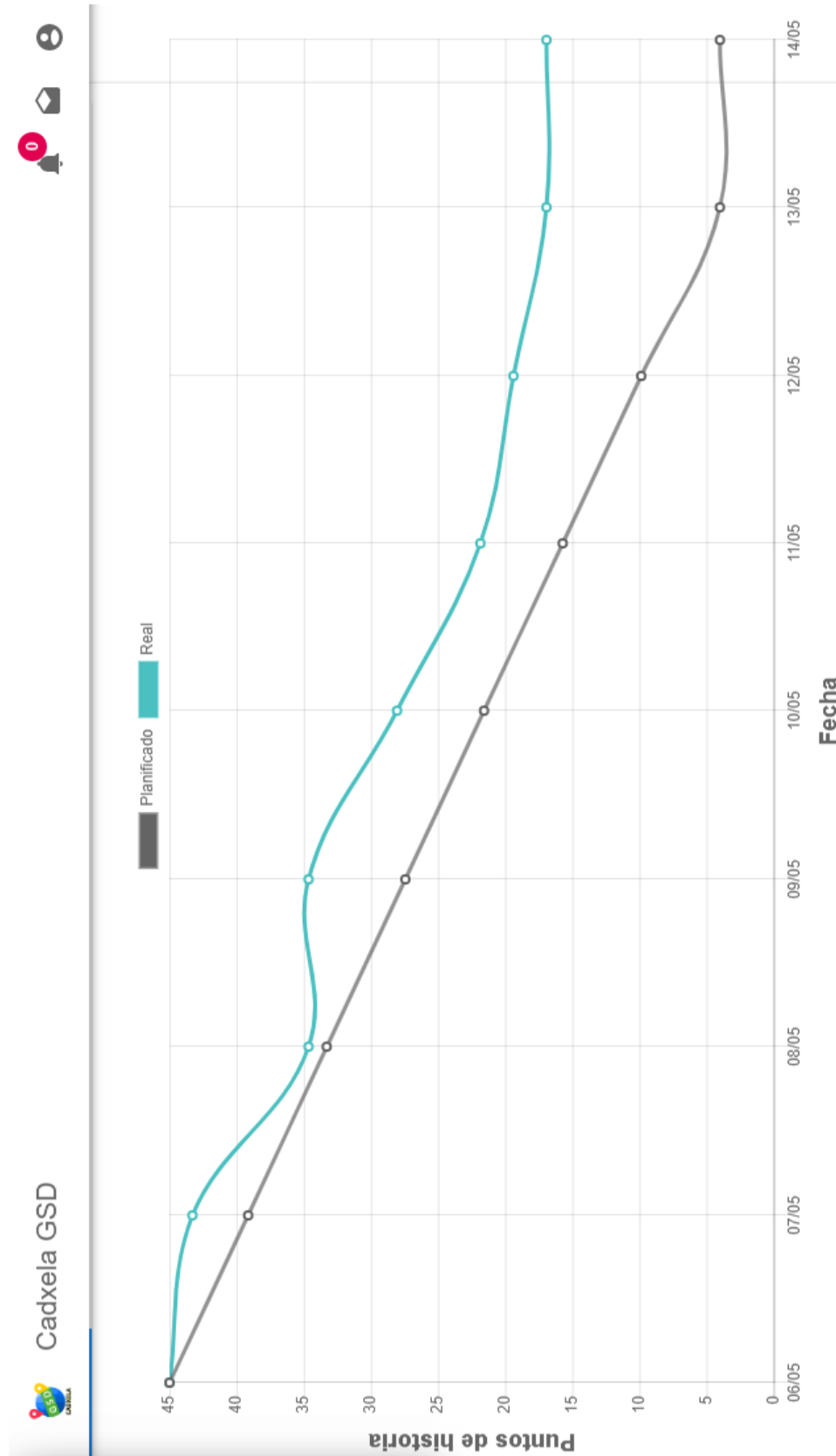


Figura 4.13 Desempeño del equipo C en la segunda ronda de proyectos.

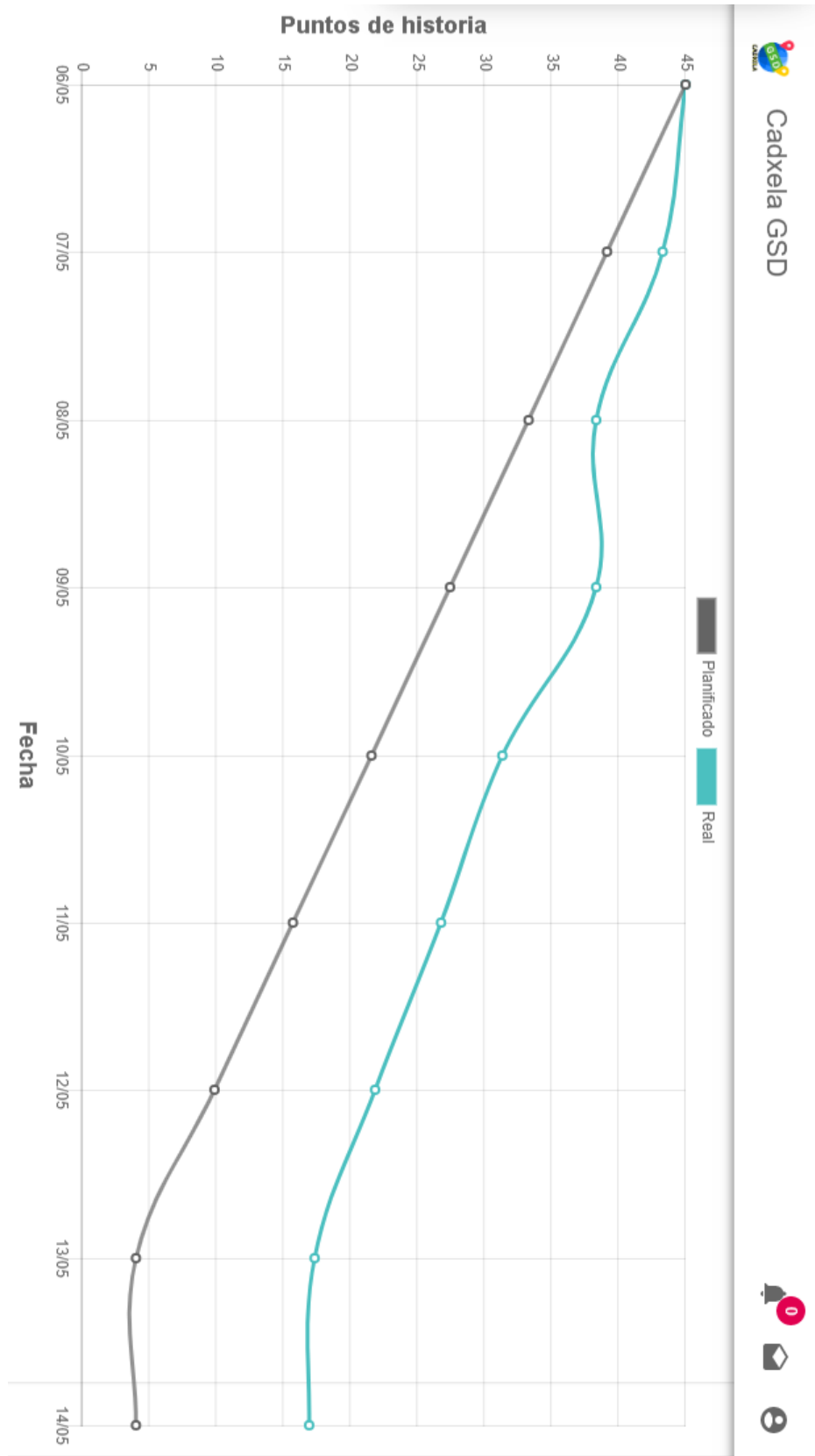


Figura 4.14 Desempeño del equipo D en la segunda ronda de proyectos.

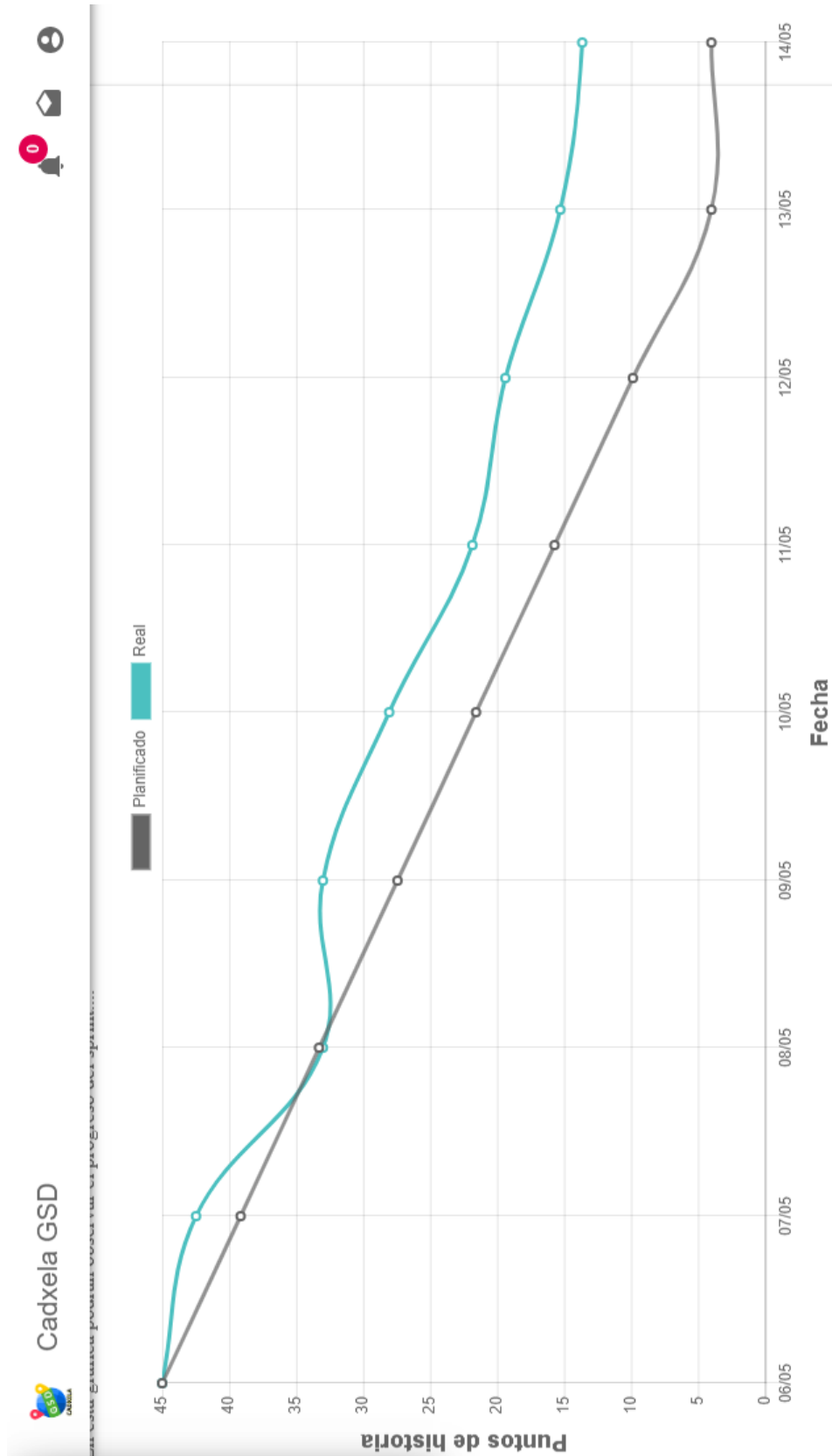


Figura 4.15 Desempeño del equipo E en la segunda ronda de proyectos.

Como se puede observar en las figuras, los equipos no pudieron terminar los proyectos de acuerdo con las estimaciones realizadas (línea gris). Concretamente, se puede afirmar que el progreso del Equipo A fue bastante inestable, puesto que hubo momentos en los cuales el equipo se acercó un poco a sus estimaciones, para luego dejar de trabajar y no terminar el proyecto. Por otro lado, se reportó un total de doce reuniones durante todo el proyecto en las cuales, en muchas ocasiones, se contó solamente con la participación de dos o a lo más tres integrantes del equipo (véase Figura 4.16).

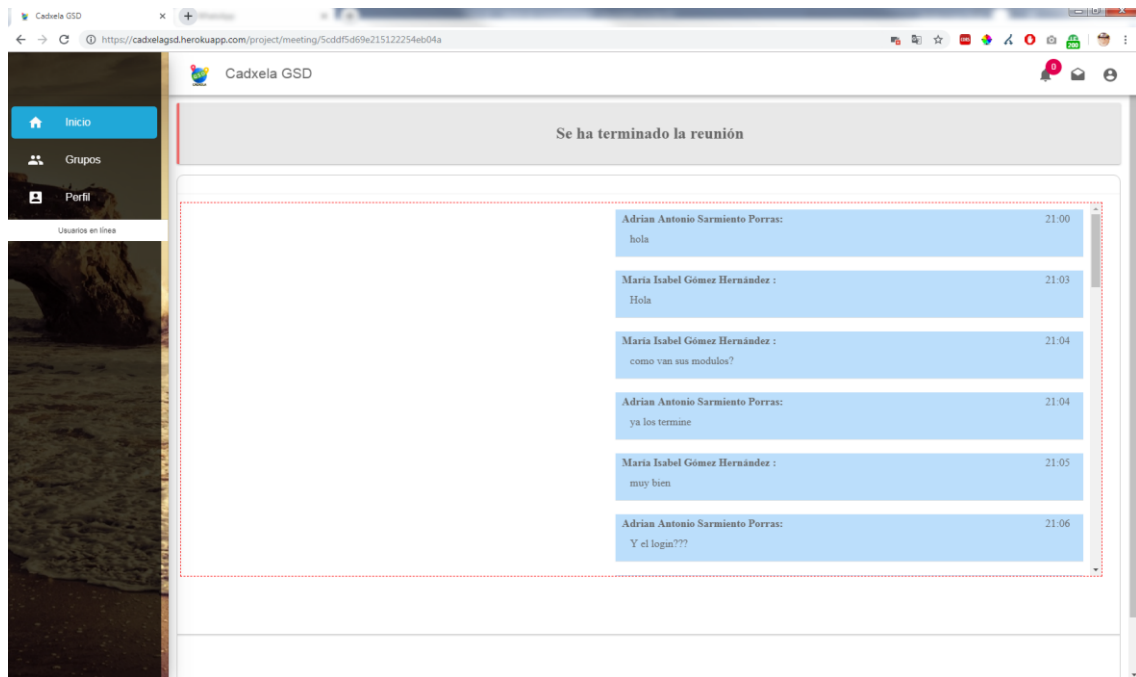


Figura 4.16. Ejemplo de una reunión donde participaron solamente dos estudiantes

Por otro lado, el Equipo E obtuvo los mejores resultados, puesto que su progreso a lo largo del proyecto estuvo bastante más cercano a las estimaciones realizadas inicialmente. En este sentido, se puede asumir que tanto el líder del proyecto como los otros integrantes del equipo llevaron a cabo un mejor trabajo de equipo, lo que les permitió obtener mejores resultados. Si bien no fue posible que este equipo terminara el proyecto, de acuerdo con su planificación inicial, sí se observa una mejora considerable en comparación con la ronda anterior de proyectos. Dicho equipo reportó un total de catorce reuniones en las cuales, al igual que en los otros equipos, en muchas ocasiones no se contó con la participación de todo el equipo. No obstante, el resto de los integrantes estuvieron pendiente de las conversaciones de sus compañeros con el objetivo de realizar las diferentes actividades asignadas.

En resumen, en esta segunda ronda de proyectos se obtuvieron mejores resultados en el desempeño de los equipos y la organización del trabajo, por lo cual se puede decir que los estudiantes con el rol de líder de proyecto entendieron y llevaron a cabo de mejorar manera sus actividades. Además, se presentaron menores dificultades en la implementación de las técnicas ágiles. Sin embargo, se detectaron problemas en cuanto al trabajo de equipo, sobre todo en la planeación y asignación de tareas puesto que la mayor parte del trabajo de planeación se delegó al líder de proyecto.

4.1.3.3.3. Tercera ronda de proyectos

Para esta última ronda de proyectos se reorganizaron los equipos y se asignaron nuevos proyectos. En estos últimos proyectos, los estudiantes estaban más relacionados con el enfoque de GSD, las técnicas ágiles implementadas en Cadxela GSD, y eran conscientes de la importancia del trabajo en equipo y de la comunicación, por lo cual la planificación de sus proyectos fue considerablemente más rápida que en las dos rondas anteriores y tuvieron más tiempo para la codificación de sus sistemas. Así mismo, los estudiantes ya dominaban los sistemas de control de versiones que les habían causado problemas en las rondas anteriores. Pese a esto, los equipos enfrentaron aún el siguiente problema:

- Falta de compromiso: Durante el desarrollo de los proyectos se presentaron casos en los cuales algunos estudiantes realizaban sus tareas de codificación demasiado rápido, y en cuanto las terminaban se ausentaban del proyecto, por lo que si sus compañeros tenían alguna duda acerca de su trabajo debían esperar a que éstos se reintegraran a las reuniones.

Por otra parte, los equipos mejoraron su trabajo en equipo y comunicación, debido a que se reportaron más reuniones de trabajo y en cada una de ellas participaban, por lo general, todos los miembros del equipo y, a diferencia de la segunda ronda, en estas reuniones se pudo identificar que la mayoría de los participantes aportaba ideas para el desarrollo del proyecto en lugar de delegar toda la responsabilidad al líder de proyecto. En adelante se muestran las figuras que ilustran el desempeño de los equipos en la última ronda de proyectos. De esta manera se puede apreciar que se obtuvieron mejores resultados en comparación con las rondas anteriores, siendo que en esta ronda incluso el equipo con peor desempeño (Equipo A) pudo entregar la mayoría de las funcionalidades del producto. Sin embargo, el comportamiento de dicho equipo fue inestable, puesto que en algunos momentos se acercó a los valores estimados, sobre todo al inicio del proyecto, y después se alejó sustancialmente de su estimación. Por otro lado, se reportó un total de catorce reuniones a lo largo del proyecto, en las cuales se pudo identificar que la mayoría de los miembros del equipo participaban y discutían sobre sus tareas. Además, el equipo presentó una mínima cantidad de problemas de coordinación sobre las tareas que les fueron asignadas.

Por último, el Equipo E que obtuvo los mejores resultados pudo entregar todas las funcionalidades del proyecto en el tiempo establecido por sus estimaciones. Dicho equipo reportó un total de dieciséis reuniones a lo largo del proyecto, en las cuales se contó con la participación de todos los miembros del equipo.

Así pues, para esta tercera ronda se mejoró sustancialmente el desempeño de los equipos, ya que todos pudieron entregar los sistemas solicitados y, además, mejoraron su trabajo en equipo y la comunicación fue más fluida entre cada uno de los participantes. Así mismo, presentaron dificultades mínimas en cuanto a los aspectos técnicos del desarrollo.

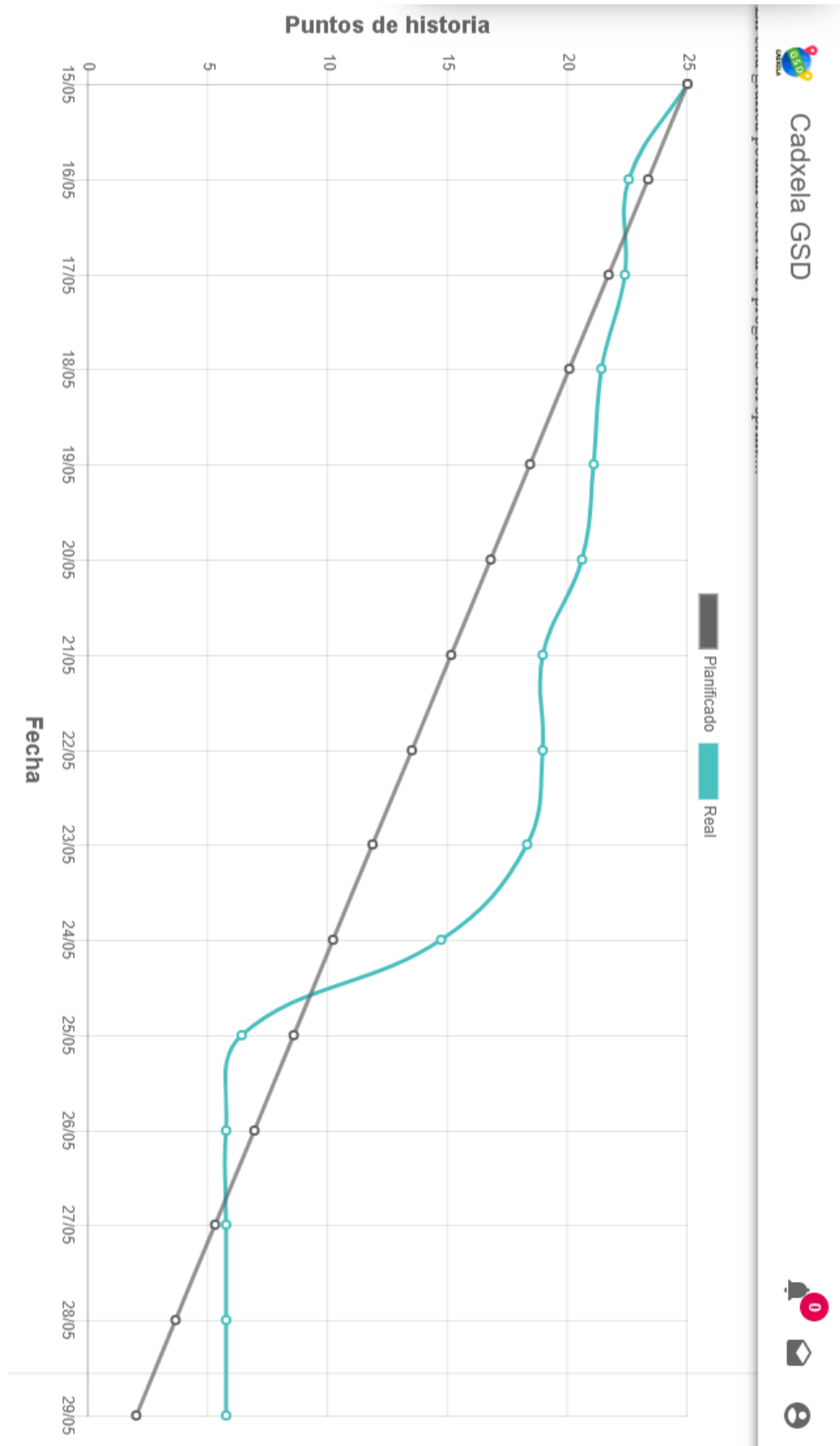


Figura 4.17 Desempeño del equipo A en la tercera ronda de proyectos.

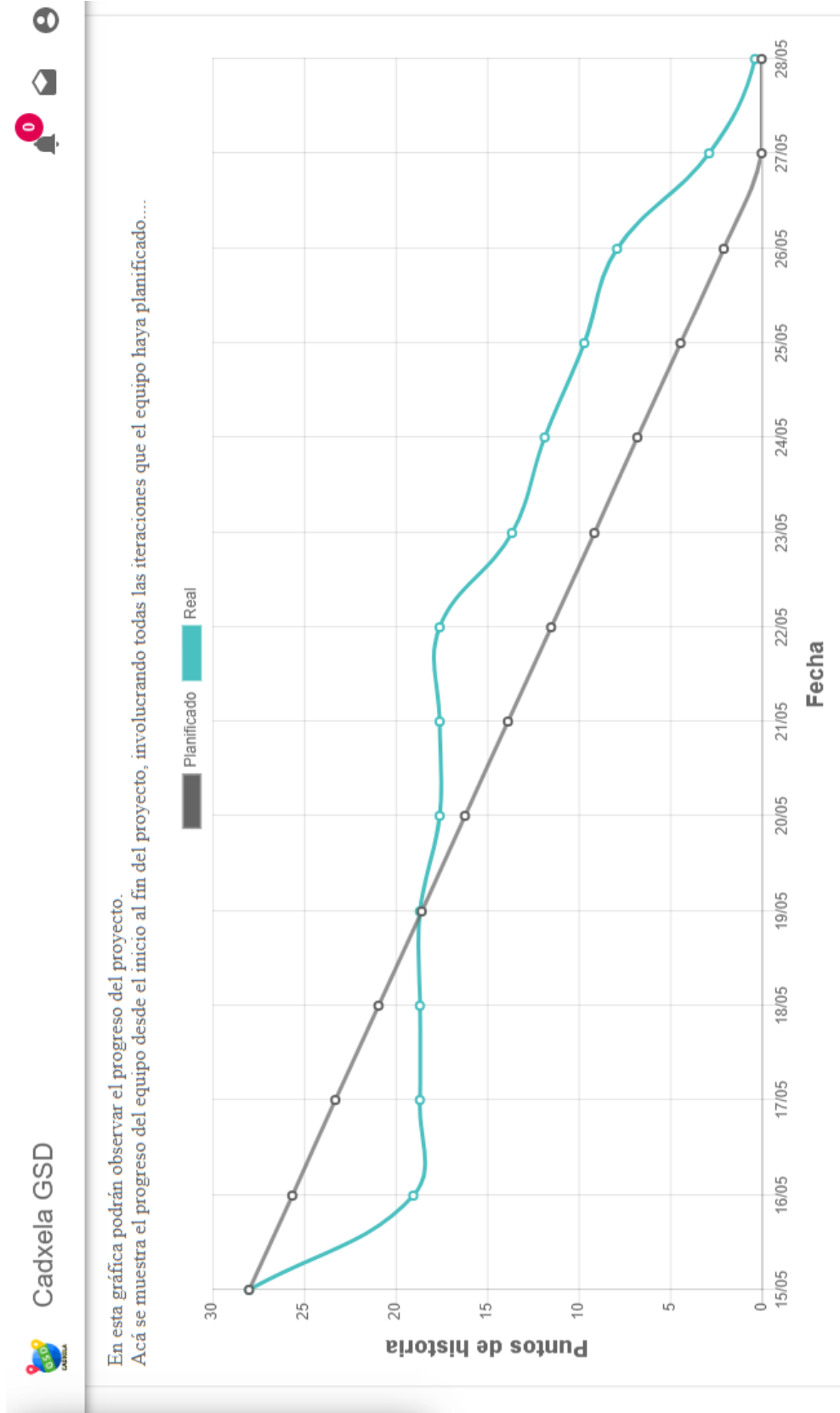


Figura 4.18 Desempeño del equipo B en la tercera ronda de proyectos.

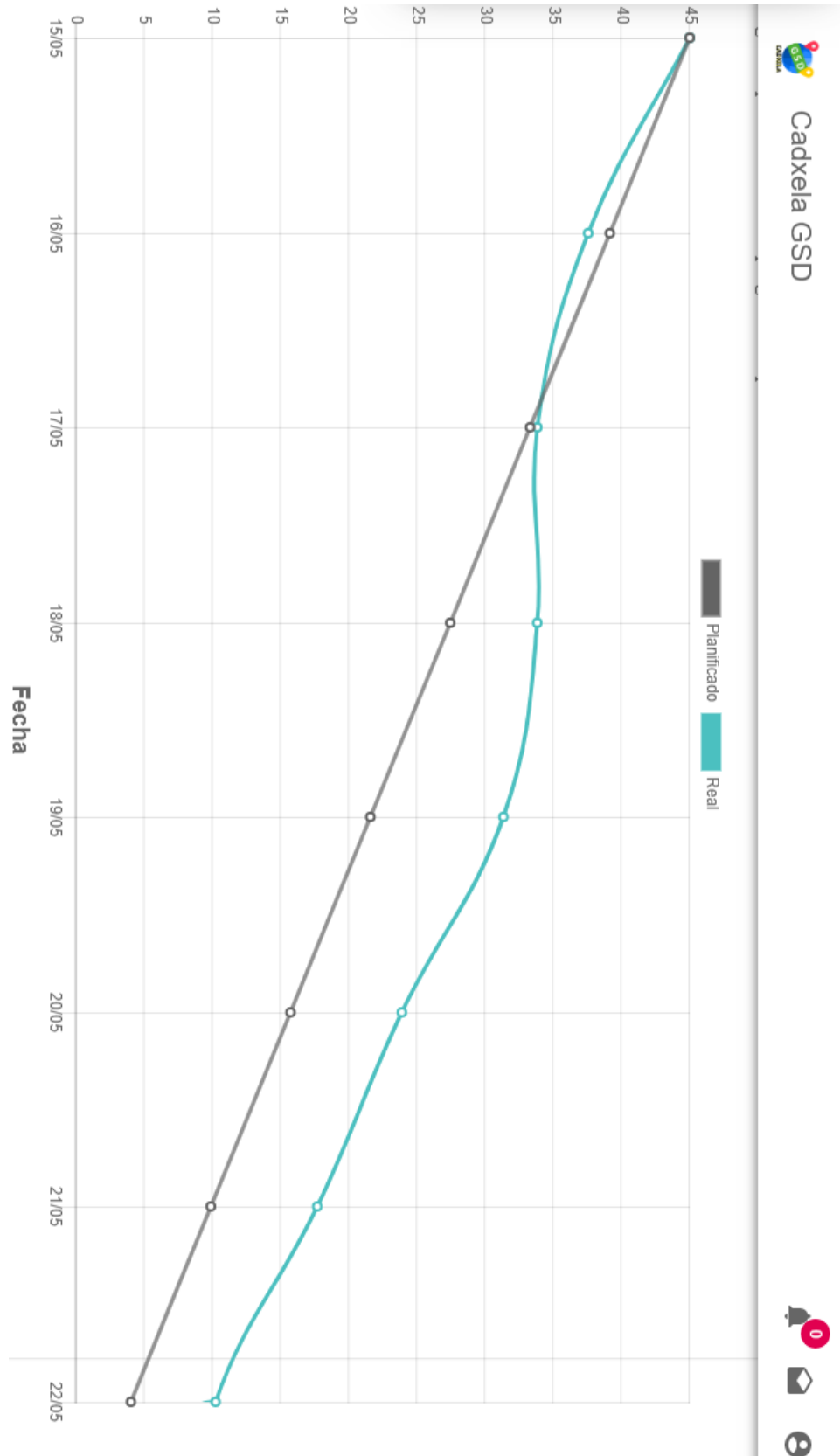


Figura 4.19 Desempeño del equipo C en la tercera ronda de proyectos.

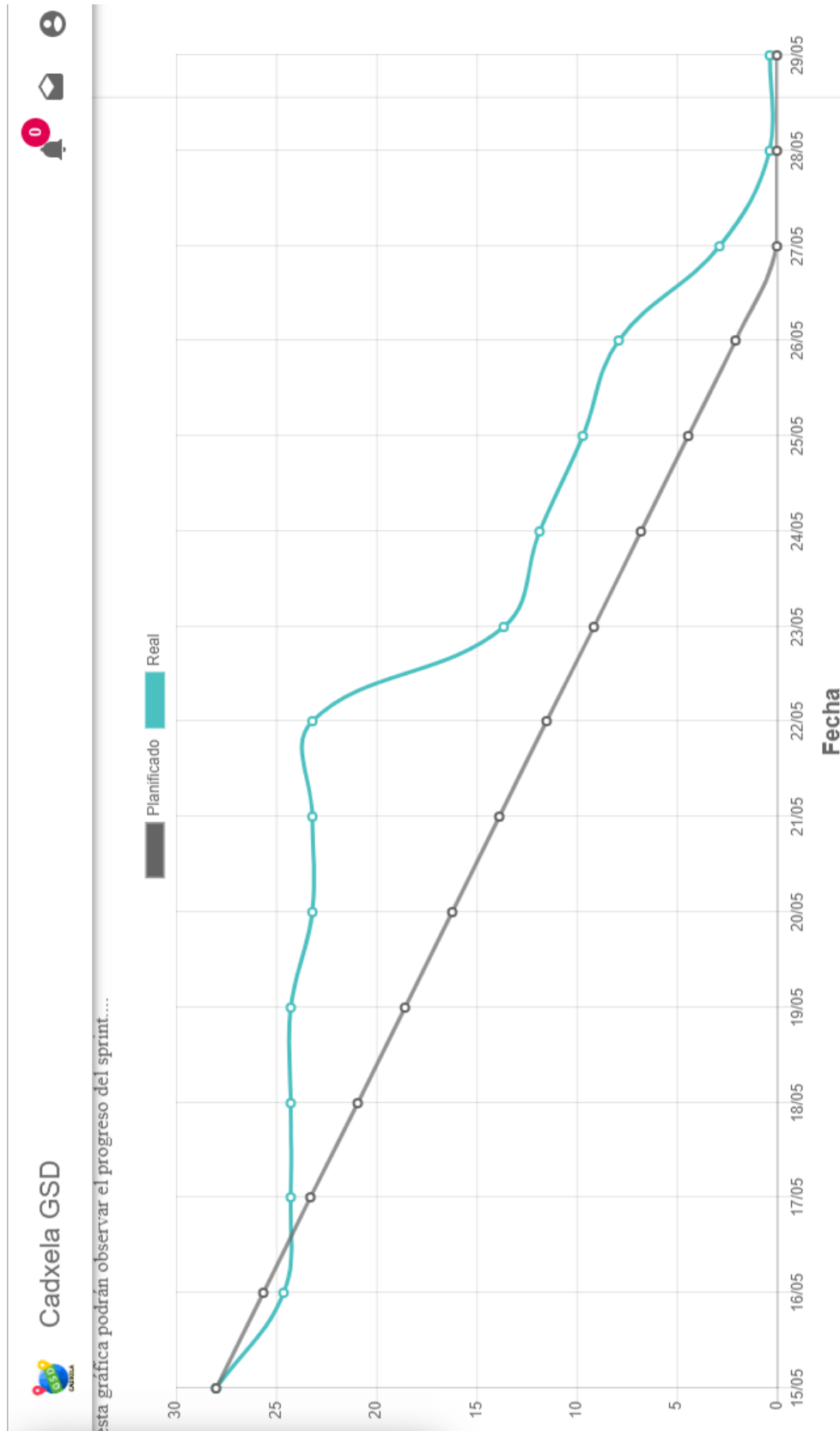


Figura 4.20 Desempeño del equipo D en la tercera ronda de proyectos.

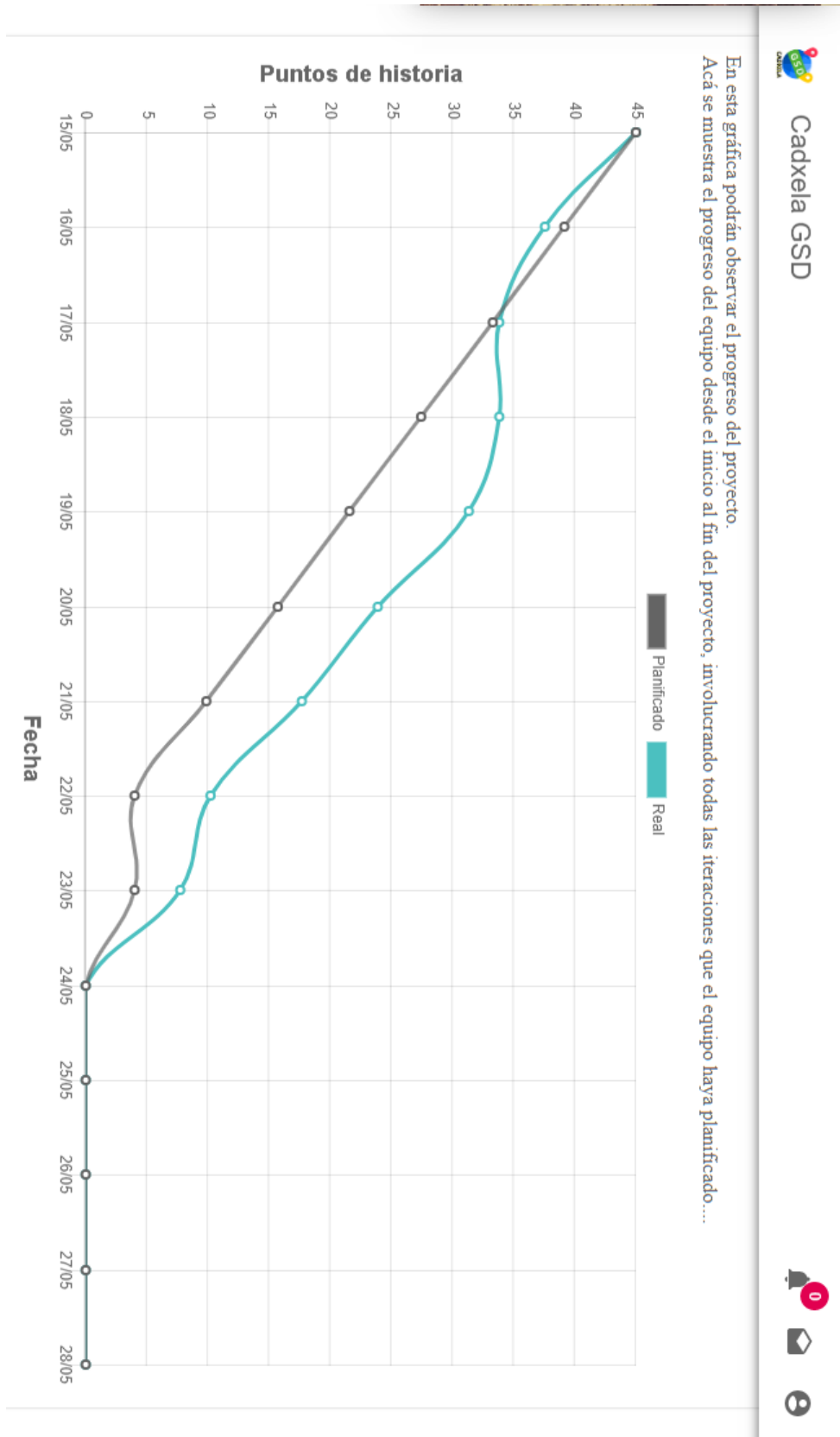


Figura 4.21 Desempeño del equipo E en la tercera ronda de proyectos.

4.1.4. Análisis de los resultados obtenidos

Como se mencionó anteriormente, se realizó una pre-evaluación de los estudiantes una vez terminado el curso teórico y antes de iniciar el desarrollo de los proyectos, con el fin de determinar el nivel de conocimiento de los estudiantes con respecto al GSD antes de la aplicación práctica de los conceptos a través de Cadxela GSD. Posteriormente, cuando se terminó la ejecución de los proyectos se aplicó una post-evaluación a los mismos estudiantes. El análisis comparativo de los resultados obtenidos con ambas evaluaciones permitió identificar si hubo una mejora con relación al conocimiento del tema después del curso. Adicionalmente, se requirió que los estudiantes respondieran una encuesta con la finalidad de evaluar su satisfacción y motivación con respecto a las actividades que habían realizado, y por ende evaluar la efectividad del marco de referencia y Cadxela GSD con respecto a la enseñanza del GSD y en términos de dos habilidades suaves: trabajo en equipo y comunicación. Por lo tanto, las siguientes secciones analizan los resultados obtenidos durante la experimentación.

4.1.4.1. Evaluación del aprovechamiento académico

Para evaluar el rendimiento académico de los estudiantes se definió un cuestionario de 20 reactivos para determinar su nivel de conocimiento sobre los temas vistos en el curso teórico. Concretamente, acerca de los conceptos del GSD, los problemas y retos presentes en éste, y las técnicas ágiles que regularmente se utilizan en la gestión de un proyecto de este tipo. Dicho cuestionario fue presentado en la Tabla 11 de la sección 4.1.2.

Dicho cuestionario fue aplicado a los estudiantes al final del curso teórico y antes de la ejecución de los proyectos de GSD, y al final de la realización de los proyectos a través de Cadxela GSD, como se aprecia en la Figura 4.22.

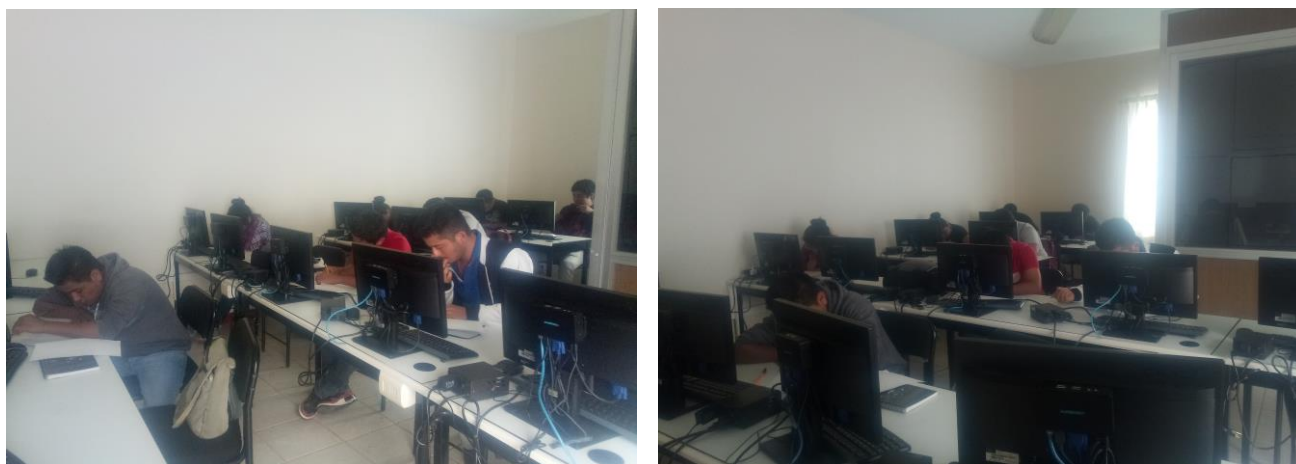


Figura 4.22. Estudiantes respondiendo la pre-evaluación (imagen izquierda) y la post-evaluación (imagen derecha)

Los resultados obtenidos por los estudiantes se muestran en la Tabla 21, en donde TC indica el total de respuestas correctas obtenidas para ese reactivo en particular, TI representa el total de respuestas incorrectas, y P muestra el promedio de las respuestas obtenidas. Se puede apreciar que los resultados obtenidos por los estudiantes en la pre-evaluación no fueron satisfactorios, ya que el promedio de las respuestas fue de 0.6. Lo anterior nos indica que probablemente los estudiantes no prestaban la atención necesaria en la clase, por lo cual no pudieron retener los conceptos. En particular se identificaron problemas en la definición de los conceptos teóricos del GSD relacionados con las primeras tres preguntas de tipo abierto, en las que se tenía que definir al GSD,

los beneficios del enfoque, y a los equipos virtuales, obteniendo un total de 10, 12, y 6 respuestas correctas respectivamente. También se identificaron problemas en las preguntas P9, que trata acerca de los problemas ocasionados por el concepto de las tres distancias, y P20, que hace alusión a la diversidad cultural. Por lo tanto, se puede inferir que a los estudiantes les costó trabajo retener los conceptos teóricos impartidos durante el curso.

No obstante, la post-evaluación arrojó una mejora sustancial en cuanto al aprovechamiento académico al obtener un promedio de 0.8 para todas las respuestas. En los resultados se puede observar que a través de la práctica los estudiantes comprendieron de mejor manera el concepto del GSD, los beneficios de éste, y aprendieron a identificar las características de los equipos virtuales, debido a que se obtuvieron un total de 16, 17, y 15 respuestas correctas respectivamente, lo cual contrasta con los resultados de la pre-evaluación. Así mismo, la pregunta P9 obtuvo mejores resultados, por lo tanto, se puede pensar que a través del desarrollo de los proyectos y de experimentar las tres distancias en un entorno real, los estudiantes pudieron comprender mejor los conceptos. Además, la pregunta P20 obtuvo un mejor resultado sugiriendo así que la práctica permitió a los estudiantes conocer la diversidad cultural. Para finalizar, las preguntas P11 a la P16, relacionadas con las técnicas ágiles, obtuvieron también mejoras en la post-evaluación. Si bien éstas son mínimas, se observó que se reforzaron los conceptos a través de la práctica. Concretamente, la pregunta P14, relacionada con la definición de las historias de usuario, obtuvo mejores resultados después de la experimentación aumentando de 16 respuestas correctas a 18. Igualmente, la respuesta P15, relacionada con la unidad de medida utilizada para las estimaciones, incrementó el número de respuestas correctas de 12 a 17.

Tabla 21. Resultados de la evaluación de conocimientos.

Pregunta	Pre-evaluación			Post-evaluación		
	TC	TI	P	TC	TI	P
P1	10	10	0.5	16	4	0.8
P2	12	8	0.6	17	3	0.85
P3	6	14	0.3	15	5	0.75
P4	13	7	0.65	15	5	0.75
P5	13	7	0.65	16	4	0.8
P6	13	7	0.65	15	5	0.75
P7	18	2	0.9	18	2	0.9
P8	10	10	0.5	16	4	0.8
P9	8	12	0.4	16	4	0.8
P10	15	5	0.75	15	5	0.75
P11	14	6	0.7	15	5	0.75
P12	15	5	0.75	16	4	0.8
P13	13	7	0.7	16	4	0.8
P14	16	4	0.75	18	2	0.9
P15	12	8	0.6	17	3	0.85
P16	16	4	0.8	16	4	0.8
P17	13	7	0.65	17	3	0.85

Pregunta	Pre-evaluación			Post-evaluación		
	TC	TI	P	TC	TI	P
P18	9	11	0.45	16	4	0.8
P19	9	11	0.45	15	5	0.75
P20	7	13	0.35	15	5	0.75
Promedio general			0.6			0.8

En resumen, se puede apreciar que a través del desarrollo de los proyectos en Cadxela GSD los estudiantes reforzaron los conceptos teóricos del GSD. Específicamente aquellos que tienen que ver con la definición del enfoque, los beneficios que éste pretende aportar a las empresas, y la definición de los equipos virtuales que participan en el desarrollo. Con relación a los equipos virtuales, se puede deducir que el hecho de trabajar en un entorno real de GSD y ser parte de un equipo virtual, sirvió para que los estudiantes comprendieran los conceptos teóricos de mejor manera. Además, el trabajar en proyectos de GSD fomentó en los estudiantes la comprensión tanto del GSD como de los beneficios que éste aporta. Por otro lado, los estudiantes se enfrentaron a los problemas ocasionados por las tres distancias (geográfica, temporal, y cultural), motivo por el cual pudieron aprender acerca de los problemas que éstas ocasionan. Para finalizar, se afirma que esta práctica también fomentó en los estudiantes el entendimiento de la diversidad cultural, sobre todo con los aspectos relacionados a las diferentes formas de trabajo y de actuar de sus compañeros. De ahí que se afirme que Cadxela GSD sirve de apoyo en la enseñanza de los conceptos teóricos del GSD.

4.1.4.2. Evaluación del rendimiento de los estudiantes en los proyectos

El rendimiento de los estudiantes en el desarrollo de los proyectos y sus habilidades de trabajo en equipo y comunicación fueron evaluadas para todos ellos con base en su desempeño, tanto individual como de equipo, en los proyectos que les fueron asignados. Es importante destacar que al inicio del caso de estudio, los estudiantes no tenían conocimiento alguno sobre el enfoque de GSD y no habían trabajado en proyectos reales de software con personas ubicadas en otro sitio geográfico, es decir, tenían nulo conocimiento y experiencia sobre el GSD.

Durante la primera ronda de proyectos se obtuvo un bajo rendimiento de los participantes. Como se expuso anteriormente, ningún equipo fue capaz de terminar el proyecto que le fue asignado. Por lo cual, el trabajo en equipo fue limitado e inclusive nulo. De igual manera, se presentaron diversos problemas para organizar el trabajo y los horarios en los cuales los estudiantes trabajarían. Debido a esto, se tuvieron retrasos en la planificación, lo que impidió que los equipos tuvieran el tiempo suficiente para trabajar en la codificación de los sistemas. A modo de ejemplo, la Figura 4.23 muestra un ejemplo de un proyecto entregado por un equipo, en donde se puede apreciar que no se terminaron las funcionalidades del mismo.

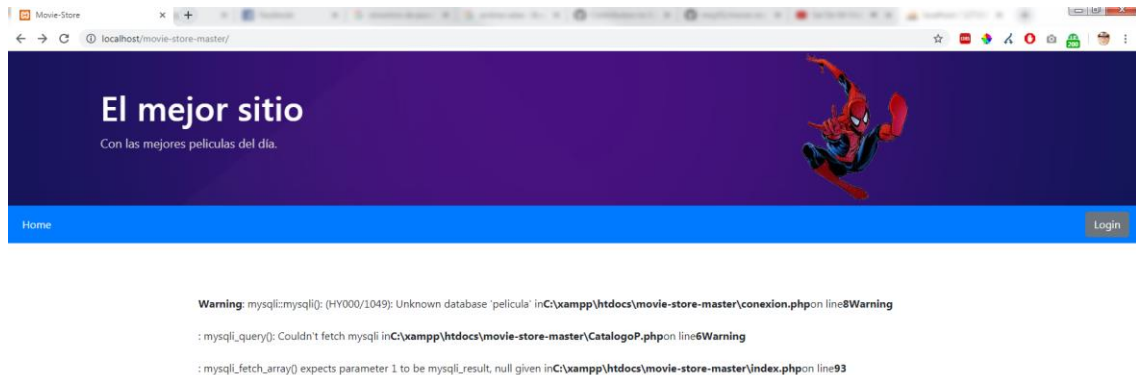


Figura 4.23. Ejemplo de proyecto incompleto entregado en la primera ronda de proyectos

Por otro lado, desde el punto de vista individual, se pudo identificar que ningún estudiante dedicó el tiempo necesario para trabajar en el proyecto, ni realizaron el esfuerzo requerido para comunicarse con sus compañeros, ya que no se pudieron identificar suficientes interacciones tanto grupales (i.e., reuniones formales) como individuales (i.e., a través de comunicación informal). Al respecto, la Figura 4.24 muestra un ejemplo de una reunión llevada a cabo por un equipo, en la cual se aprecia la falta de comunicación y trabajo en equipo.

Igualmente, el trabajo llevado a cabo por los estudiantes con el rol de líder de proyecto fue deficiente, puesto que no fueron capaces de llevar a buen puerto el desarrollo del proyecto y, en general, no estuvieron involucrados en el trabajo.

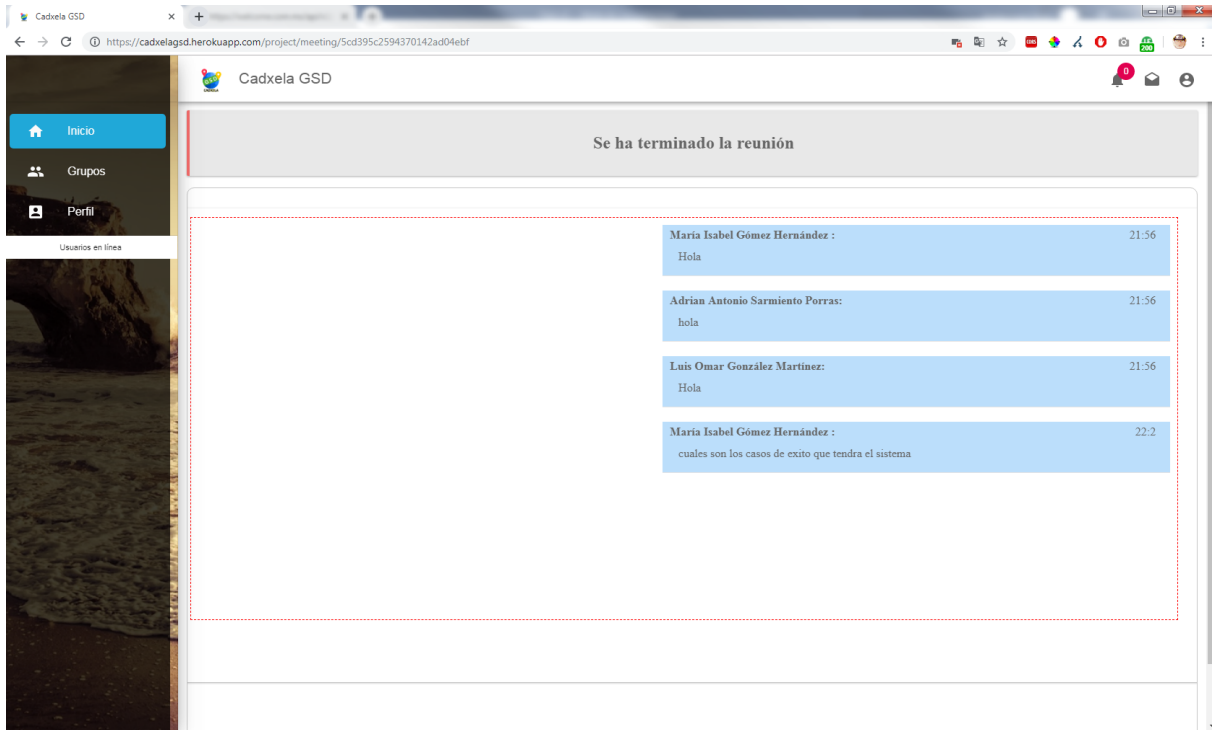


Figura 4.24. Ejemplo de un problema de comunicación y trabajo en equipo suscitado en la primera ronda

Por otro lado, para la segunda ronda de proyectos el rendimiento de los equipos mejoró considerablemente, puesto que se entregaron sistemas funcionales que, si bien no se terminaron en tiempo y forma, mostraron una mejora en cuanto al producto final entregado (véase Figura 4.25).

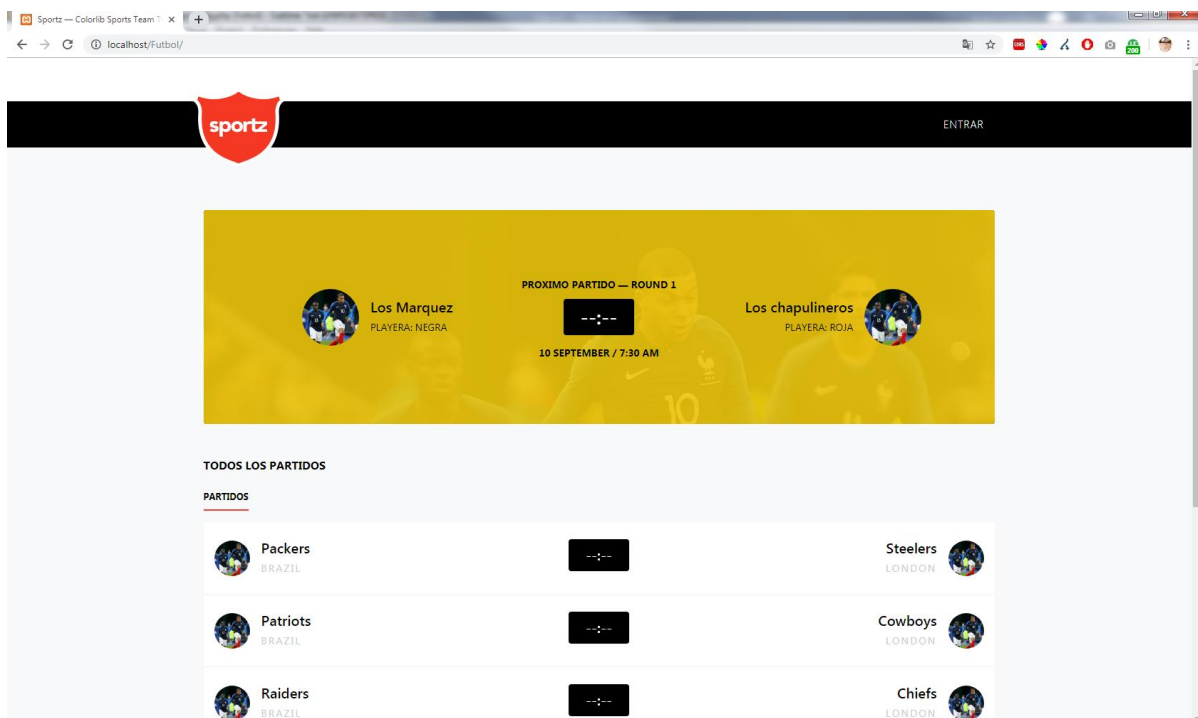


Figura 4.25. Ejemplo de proyecto entregado en la segunda ronda de proyectos

En lo que respecta al trabajo en equipo y la comunicación también se reportaron mejoras sustanciales. Es decir, las reuniones formales y la participación de los estudiantes se incrementaron (véase Figura 4.26). Sin embargo, no se hizo uso de la comunicación informal por lo que toda la comunicación fue meramente formal, para ello los estudiantes establecieron un horario de trabajo y si algún participante no podía asistir debía de leer los temas tratados en la reunión para estar al corriente con sus compañeros. En síntesis, el trabajo en equipo y la comunicación fueron fundamentales para que los productos entregados cumplieran con la mayoría de las funcionalidades requeridas.

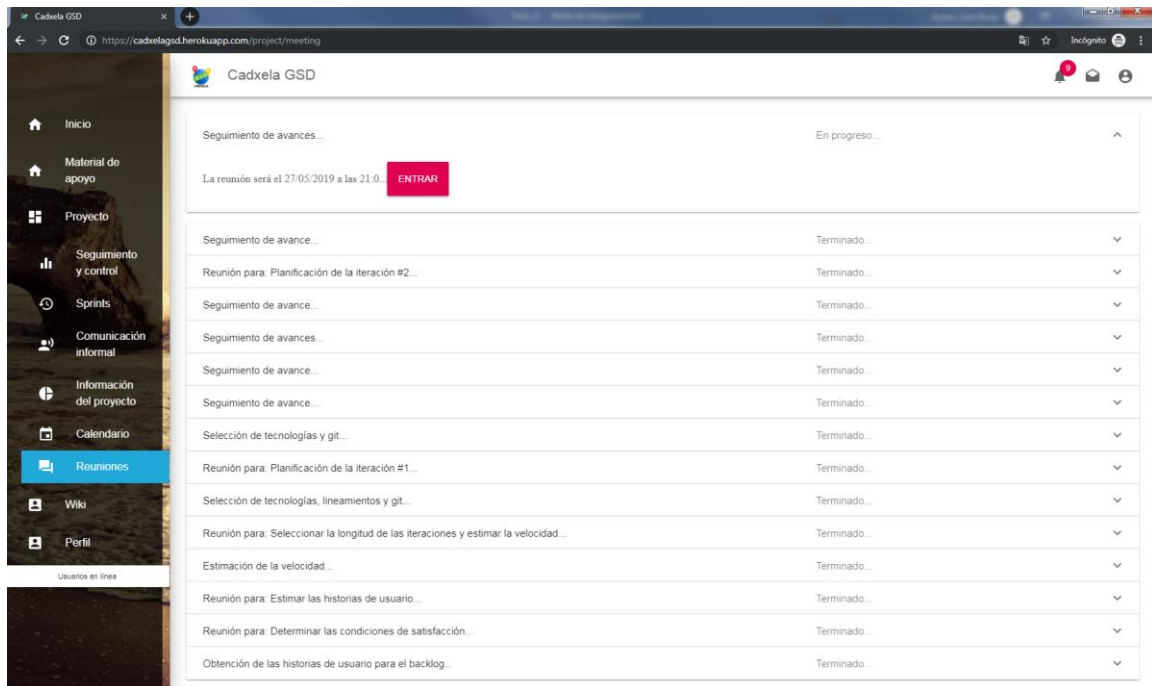


Figura 4.26. Tablero de reuniones conducidas por un equipo durante la segunda ronda

Por último, durante la tercera ronda de proyectos se continuó con la tendencia sobre la mejora del rendimiento de los equipos. En este caso, los productos ya fueron entregados cumpliendo con todas las funcionalidades requeridas para cada proyecto. En este contexto, el trabajo de los jefes de proyecto mejoró considerablemente para que los equipos terminaran de mejor manera los proyectos. La Figura 4.27 muestra el ejemplo de un sistema entregado en esta tercera ronda.

El trabajo en equipo y la comunicación también mejoró sustancialmente con respecto a los proyectos anteriores, en esta ronda los integrantes de los equipos aportaban ideas para la solución de problemas y no solamente dejaban esta responsabilidad al líder de proyecto. Además, se reportaron un mayor número de reuniones formales. No obstante, la comunicación informal fue nula. A modo de ejemplo, la Figura 4.28 muestra el ejemplo de una reunión en donde se contó con la participación de los integrantes para la resolución de los problemas relacionados con el proyecto.

En resumen, el rendimiento de los equipos tanto a nivel grupal como individual mejoró sustancialmente conforme se realizaban los proyectos. En los primeros proyectos los estudiantes no conocían el enfoque del GSD ni las técnicas que aplicarían en Cadxela GSD, pero según se avanzaba en la realización de otros proyectos fueron adquiriendo la experiencia necesaria para cada vez realizar un mejor trabajo. Así mismo, la comunicación y el trabajo en equipo fueron mejorando y esto les permitió a los equipos obtener mejores resultados en la entrega de los proyectos. De ahí que se pueda inferir que Cadxela GSD permitió que los estudiantes mejoraran su desempeño en el

desarrollo de proyectos de GSD, al mismo tiempo que mejoraron sus habilidades de trabajo en equipo y comunicación.

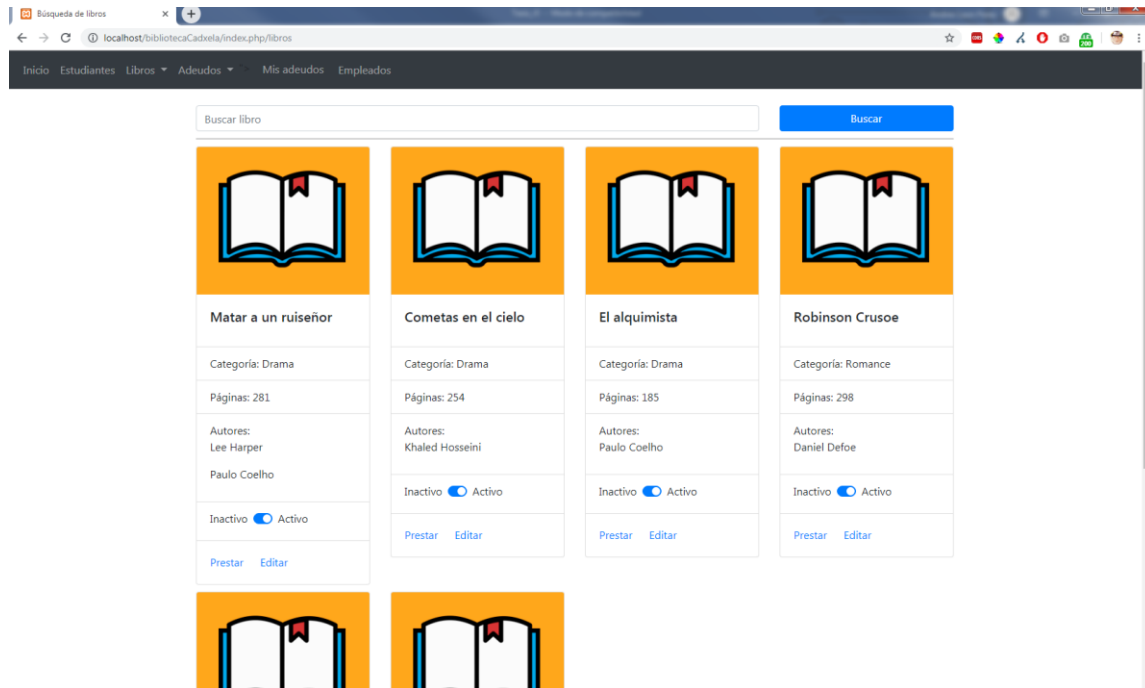


Figura 4.27. Ejemplo de proyecto entregado en la tercera ronda

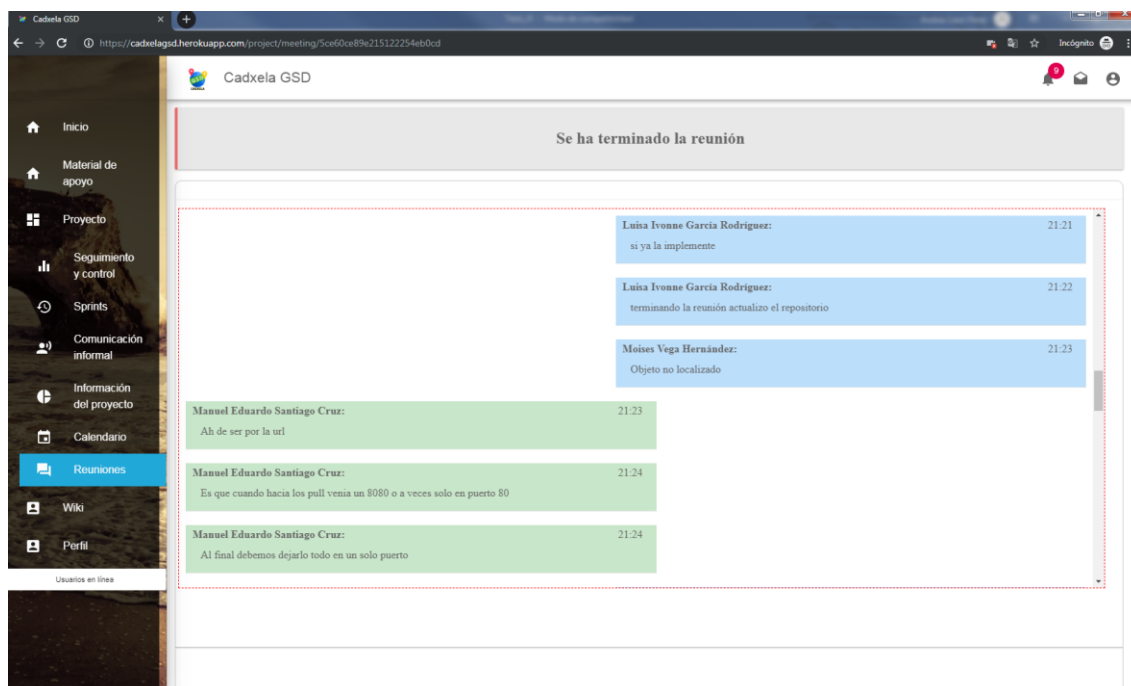


Figura 4.28. Ejemplo de una reunión exitosa en la tercera ronda de proyectos

4.1.4.3. Evaluación de la motivación de los estudiantes

La encuesta aplicada a los participantes estuvo conformada por 15 ítems para evaluar la motivación de los estudiantes con respecto al marco de referencia y el uso de Cadxela GSD durante los proyectos realizados. La Tabla 22 muestra los resultados obtenidos en la encuesta enfocada a valorar la motivación de los estudiantes.

Tabla 22. Resultados de la encuesta para evaluar la motivación de los estudiantes.

Ítem	TA	A	D	TD	M	DE
1. Encontré que la práctica realizada representó un reto estimulante para mí	6	10	4	0	3.1	0.7
2. Durante las clases teóricas me sentí motivado por aplicar los conceptos teóricos en un entorno práctico	2	15	3	0	2.9	0.5
3. Durante la ejecución de los proyectos me sentí motivado por terminarlos en tiempo y forma	7	9	4	0	3.1	0.7
4. Durante la ejecución de los proyectos me sentí motivado por aprender sobre otras formas de trabajo de mis compañeros	6	12	2	0	3.2	0.6
5. Me sentí motivado por colaborar con personas de diferentes sitios con los cuales no había trabajado anteriormente	5	12	3	0	3.1	0.6
6. A través del desarrollo de los proyectos de GSD me di cuenta de la importancia de capacitarme en este enfoque	9	9	2	0	3.3	0.6
7. A través del desarrollo de los proyectos y el uso de Cadxela GSD me di cuenta de la importancia del trabajo en equipo y la comunicación.	4	13	3	0	3	0.6
8. Durante la práctica me sentí motivado por gestionar y controlar el proyecto que me fue asignado	3	14	3	0	3	0.5
9. Disfruté el realizar los proyectos en el entorno del GSD que fue establecido	0	15	5	0	2.7	0.4
10. El enfoque educativo utilizado para realizar la práctica me sirvió para ganar confianza sobre lo que aprendí en las clases teóricas	5	11	4	0	3	0.6
11. Fui capaz de cumplir con la tarea que se me asignó a pesar de todos los inconvenientes que se presentaron durante el desarrollo de los proyectos	3	17	0	0	3.1	0.3
12. Quedé satisfecho con mi desempeño como líder de proyecto	0	11	7	2	2.4	0.6
13. Quedé satisfecho con mi desempeño como programador y miembro de un equipo de GSD	0	15	5	0	2.7	0.4
14. Estoy convencido que colaboré eficazmente en el desarrollo de los proyectos que me fueron asignados	4	14	2	0	3.1	0.5
15. Me agradó la forma de trabajo que se llevó a cabo en el curso en cada uno de los equipos en los que participé	4	10	4	2	2.8	0.8

En general, los estudiantes afirman que se sintieron motivados durante la ejecución del caso de estudio. Particularmente, el 85% de los participantes se sintió motivado por trabajar con personas de diferentes sitios y compartir y aprender acerca de nuevas ideas y formas de trabajo, como se aprecia en las respuestas obtenidas para el ítem “5. *Me sentí motivado por colaborar con personas de diferentes sitios con los cuales no había trabajado*”. Así mismo, resalta el hecho de que el uso de Cadxela GSD permitió que los estudiantes comprendieran la importancia del trabajo en equipo y la comunicación, como se aprecia en el ítem “7. *A través del desarrollo de los proyectos y el uso de Cadxela GSD me di cuenta de la importancia del trabajo en equipo y la comunicación*”, ya que

estas habilidades forman parte de los objetivos de esta tesis. Por otra parte, el 100% de los estudiantes se sintió satisfecho con el trabajo que realizó en cada uno de los proyectos ya que, como se aprecia en el ítem “11. *Fui capaz de cumplir con la tarea que se me asignó a pesar de todos los inconvenientes que se presentaron durante el desarrollo de los proyectos*”, los estudiantes consideran que pudieron llevar a cabo todas las tareas que les fueron asignadas.

Con relación a la gestión de los proyectos, se observó que los estudiantes no se sintieron satisfechos con el trabajo que realizaron, ya que para el ítem “12. *Quedé satisfecho con mi desempeño como líder de proyecto*” se obtuvo una media de respuestas de 2.4 lo cual indica que el 35% de los estudiantes estuvo en desacuerdo con la afirmación, mientras que el 10% estuvo totalmente en desacuerdo. Esto se puede deber a que, como se mencionó anteriormente, se observó que el rol de líder de proyecto requirió de un mayor esfuerzo y no fue hasta la tercera ronda de proyectos que se logró una mejora sustancial en el desempeño. Por otra parte, algunos estudiantes (25%) afirmaron que no disfrutaron el realizar la práctica en el entorno establecido, como se muestra en los resultados obtenidos para el ítem “9. *Disfruté el realizar los proyectos en el entorno del GSD que fue establecido*”. En este sentido, durante el desarrollo de los proyectos se pudo notar que para algunos estudiantes requirió un mayor esfuerzo el realizar las actividades con sus compañeros en el entorno de GSD, ya que no podían trabajar en los horarios establecidos o tenían dificultades técnicas tanto de servicio de internet, como de dominio de herramientas y lenguajes de programación. No obstante, tuvieron que adaptarse a la forma de trabajo establecida por el equipo. También es importante destacar que en el ítem “15. *Me agradó la forma de trabajo que se llevó a cabo en el curso en cada uno de los equipos en los que participé*” se presentó una variedad de respuestas, desde estudiantes que estuvieron totalmente de acuerdo, hasta aquellos que estuvieron totalmente en desacuerdo. Por lo tanto, se cree que este comportamiento se debe a que muchos estudiantes no participaron activamente en cada uno de los equipos en los que trabajaron. Peor aún, se dieron casos en los cuales los estudiantes realizaban su trabajo, pero no generaron empatía con sus compañeros, por lo cual se limitaban a trabajar, pero no colaboraban en la solución de los problemas.

En conclusión, se puede afirmar que los estudiantes se sintieron motivados por trabajar en el entorno de GSD pero, al presentarse diversos problemas como los mencionados en la sección 4.1.4.2, consideran que no pudieron disfrutar totalmente el entorno del GSD planteado. Por otro lado, los participantes pudieron comprender a través de la práctica la importancia del trabajo en equipo y la comunicación, dado a que al enfocarse en utilizar mejor estas habilidades pudieron obtener mejores resultados en los proyectos finales.

4.1.4.4. Evaluación de la satisfacción de los estudiantes

La encuesta aplicada a los estudiantes estuvo conformada por 9 ítems que evalúan su satisfacción con respecto al marco de referencia y el uso de Cadxela GSD. La Tabla 23 muestra los resultados obtenidos en la encuesta.

Tabla 23. Resultados de la encuesta para evaluar la satisfacción de los estudiantes.

Ítem	TA	A	D	TD	M	DE
1. Durante el desarrollo de los proyectos me sentí cómodo aplicando las técnicas ágiles seleccionadas a través de Cadxela GSD	6	12	1	1	3.1	0.7
2. Durante el desarrollo de los proyectos pude llevar a cabo de buena manera la comunicación con los miembros de mi equipo de trabajo a través de Cadxela GSD	2	15	3	0	2.9	0.5
3. Durante el desarrollo de los proyectos puede trabajar eficazmente con mi equipo de trabajo	0	16	4	0	2.8	0.4

Ítem	TA	A	D	TD	M	DE
4. Me sentí cómodo aplicando las técnicas seleccionadas para llevar a cabo la planeación de los proyectos en los que trabajé	6	12	2	0	3.2	0.6
5. Me sentí cómodo llevando la gestión del proyecto que me fue asignado a través de Cadxela GSD	3	14	3	0	3	0.5
6. Sentí que las clases teóricas me sirvieron para poder realizar las diferentes actividades que me fueron asignadas durante la ejecución de los proyectos	9	9	2	0	3.3	0.6
7. Colaboré de manera eficaz con mis compañeros de equipo	4	14	2	0	3	0.5
8. Conocí, y me gustaría seguir mejorando, mis habilidades para poder trabajar en entornos de GSD	8	12	0	0	3.4	0.6
9. Sin duda alguna volvería a participar en este tipo de proyectos	5	14	1	0	3.2	0.5

La información mostrada en la Tabla 23 muestra claramente que el uso de Cadxela GSD permitió que los estudiantes aplicaran correctamente las diferentes técnicas para desarrollar un proyecto en un entorno de GSD. Mejor aún, los participantes afirmaron que las clases teóricas les sirvieron para realizar sus actividades prácticas en la plataforma. Por otra parte, se puede observar a través de los ítems “2. Durante el desarrollo de los proyectos pude llevar a cabo de buena manera la comunicación con los miembros de mi equipo de trabajo a través de Cadxela GSD” y “3. Durante el desarrollo de los proyectos puede trabajar eficazmente con mi equipo de trabajo” que se presentaron diversos problemas entre los estudiantes que dificultaron el trabajo en equipo y la comunicación. Sin embargo, como ya se analizó anteriormente, se obtuvieron mejoras significativas en cuanto a estas habilidades conforme los estudiantes avanzaban en el desarrollo de los proyectos.

4.1.4.5. Evaluación de la opinión de profesores

Por último, se aplicó la encuesta de aceptación de Cadxela GSD a los cinco profesores participantes. En este sentido, la Tabla 24 muestra los resultados obtenidos con dicha aplicación.

Tabla 24. Resultados de la aplicación de la encuesta de opinión de profesores.

Ítem	TA	A	D	TD	M	DE
1. Pienso que el uso de Cadxela GSD me ayudó en el proceso de la enseñanza práctica del GSD	1	4	0	0	3.2	0.4
2. Pienso que el uso de Cadxela GSD me ayudó a preparar de mejor manera mis clases sobre el GSD	0	2	3	0	2.4	0.5
3. Pienso que el uso de Cadxela GSD permitió que mis estudiantes mejoraran sus conocimientos y habilidades relacionadas con el GSD	2	3	0	0	3.4	0.5
4. Pienso que Cadxela GSD sería una herramienta útil en mi trabajo docente	1	4	0	0	3.2	0.4
5. Pienso que el uso de Cadxela GSD permitió que mis estudiantes estén más motivados por aprender y reforzar los conceptos teóricos del GSD	0	4	1	0	2.8	0.4
6. Me resultó fácil organizar equipos y asignarles proyectos a través de Cadxela GSD	3	2	0	0	3.6	0.5
7. Me resultó fácil monitorear el progreso de los equipos a través de Cadxela GSD	3	2	0	0	3.6	0.5
8. Me resultó fácil comunicarme con mi equipo de estudiantes a través de Cadxela GSD	2	2	1	0	3.2	0.8
9. Interactuar con Cadxela GSD no requirió de mucho esfuerzo	4	0	1	0	3.6	0.8

Ítem	TA	A	D	TD	M	DE
10. Tampoco requerí de mucho esfuerzo para aprender a utilizar Cadxela GSD	2	3	0	0	3.4	0.5
11. Cuando requerí de ayuda para resolver algún problema con Cadxela GSD, conté con instrucciones para ayudarme a hacerlo	0	5	0	0	3	0
12. Cuando necesité de ayuda para resolver algún problema con Cadxela GSD, siempre hubo una persona dispuesta a ayudarme	3	2	0	0	3.6	0.5
13. Es más interesante enseñar los conceptos relacionados con el GSD siguiendo el enfoque establecido por Cadxela GSD	1	4	0	0	3.2	0.4

Los resultados mostrados anteriormente evidencian que los profesores participantes estuvieron de acuerdo en que Cadxela GSD es una herramienta que puede servir de apoyo en la enseñanza del GSD. Sin embargo, las respuestas al ítem “2. *Pienso que el uso de Cadxela GSD me ayudó a preparar de mejor manera mis clases sobre el GSD*” indicaron que el 60% de los profesores también consideró que Cadxela GSD no le fue de utilidad para preparar sus clases acerca de los temas puestos en práctica. Lo anterior se debe a que, en verdad, la plataforma no está creada para que el profesor prepare el material de su clase, sino que cuenta con material pre-cargado con contenido sugerido para que los profesores puedan mejorar su clase. Por tal motivo, se puede afirmar que probablemente la interfaz de Cadxela GSD no es lo suficientemente clara como para que los profesores consulten y utilicen el contenido pre-cargado. Pese a este punto, se observó que los profesores opinaron que Cadxela GSD es una herramienta útil y que les puede servir de apoyo para mejorar su rendimiento en la enseñanza del GSD. Por otro lado, se observa también que estos mismos profesores no requirieron de mucho esfuerzo para hacer uso de la plataforma y que tuvieron a su disposición las facilidades necesarias para utilizarla. En concreto, Cadxela GSD fue aceptada por los profesores como herramienta de apoyo en la enseñanza del GSD.

4.1.4.6. Consideraciones finales sobre los resultados obtenidos

A modo de resumen, los resultados mostrados anteriormente evidencian que los estudiantes se mostraron motivados durante la ejecución del caso de estudio, pero en general sintieron que su rendimiento personal no fue el esperado. No obstante, el desempeño de éstos fue de menos a más, por lo que se puede afirmar que obtuvieron experiencia práctica acerca de la forma de trabajo en el enfoque del GSD siguiendo los lineamientos establecidos por el marco de referencia propuesto. Aunado a lo anterior, se pudo observar una mejora significativa en los estudiantes con respecto a sus habilidades de trabajo en equipo y comunicación, ya que, si bien al principio del desarrollo de los proyectos enfrentaron diversos problemas relacionados con estas habilidades, fueron adquiriendo la experiencia necesaria para cada vez realizar un mejor trabajo. En este sentido, ambas habilidades fueron piezas clave para que en los últimos proyectos se obtuvieran mejores resultados. En cambio, el aprovechamiento académico de los estudiantes no presentó mejoras significativas, solamente se puede destacar el hecho de que a través de la práctica comprendieron el concepto del GSD, pero no se reforzaron otros conceptos importantes como la diversidad cultural, la gestión del conocimiento, y la colaboración.

Por último, la información vertida en este capítulo pretende reunir la evidencia que permita aceptar o rechazar la hipótesis planteada en el Capítulo 1 de esta tesis, la cual se estableció de la siguiente manera:

Hipótesis: *“La creación e implementación de un marco de referencia permitirá que los estudiantes desarrollen las habilidades suaves de comunicación y trabajo en equipo en entornos de GSD”*

Esta hipótesis es aceptada dado que como se presentó anteriormente, los estudiantes fueron capaces de mejorar sus habilidades de trabajo en equipo y comunicación para poder terminar los proyectos que les fueron asignados. En este sentido, en el análisis del rendimiento de los estudiantes se mostró cómo éstos fueron progresando en sus habilidades tanto de trabajo en equipo como de comunicación a lo largo del desarrollo de los diferentes proyectos. De ahí que se afirme que estas habilidades fueron fundamentales para obtener una mejora en el desempeño de los estudiantes. Es importante mencionar que esta hipótesis, que ha sido establecida en el contexto de la tesis, es aceptada para el caso de estudio establecido en la experimentación. De ninguna manera es posible generalizar que el uso del marco de referencia y Cadxela GSD proporcionará los mismos resultados en cualquier entorno de enseñanza del GSD debido a que esto requiere, de forma obligatoria, continuar con la experimentación incluyendo una muestra más significativa. No obstante, en el contexto de la experimentación en la IS se afirma que los resultados positivos sobre una efectividad parcial, mostrados en un trabajo de investigación o reporte, evidencian que es posible establecer un caso de estudio con un mayor número de participantes y proyectos (Juristo & Moreno, 2010).

4.2. Amenazas a la validez del caso de estudio

De acuerdo con Runeson y Höst (2009), la validez de un caso de estudio indica la confiabilidad de los resultados. En otras palabras, ésta indica en qué medida los resultados son verdaderos y no están influenciados por el punto de vista subjetivo de los investigadores. Para ello, se sugiere identificar un conjunto de amenazas a la validez para los casos de estudio en el campo de la investigación de la IS que sean útiles para validar los resultados presentados en esta tesis. Por lo tanto, las principales amenazas a la validez del caso de estudio presentado son las siguientes:

- **Validez interna:** Los problemas de validez interna tratan los problemas causales relacionados con los resultados presentados. Por lo tanto, es importante mencionar que ninguno de los estudiantes y profesores participantes tenían conocimiento previo acerca de la preparación del caso de estudio, por lo cual no modificaron su comportamiento al realizar las actividades prácticas y, por lo tanto, se evitó ocasionar un sesgo en los resultados. Sin embargo, no se puede asegurar que los estudiantes participantes tuvieran el mismo nivel de conocimientos en cuanto al desarrollo de software, metodologías ágiles, entre otros, en comparación con otros estudiantes que no participaron en el caso de estudio. Además, tampoco se puede confirmar que los participantes tuvieran mejores o peores habilidades para comunicarse y trabajar en equipo, o que tuvieran conocimiento previo acerca del desarrollo de software a distancia. Finalmente, puesto que el trabajo de los equipos y la evaluación de sus resultados fueron monitoreados por los profesores, no es posible asegurar que el desempeño de los equipos estuvo influenciado por el nivel de participación y/o interés de los profesores.
- **Validez de construcción:** Los problemas de validez de construcción pueden surgir de errores durante la evaluación. Para lidiar con estas amenazas, se recopilaron datos cuantitativos y cualitativos una vez que se terminaron las rondas de proyectos, con el objetivo de garantizar que los resultados fueran los mismos, independientemente de quien los analizara. Sin embargo, resulta difícil evaluar la motivación, satisfacción, y grado de aceptación de una tecnología, por lo cual, se diseñaron cuestionarios utilizando una escala de tipo Likert de 4 puntos para así obtener un valor cuantitativo de la motivación y satisfacción de los

estudiantes. Así mismo, para evaluar el grado de aceptación de Cadxela GSD se diseñó un cuestionario tomando como base los lineamientos de los modelos TAM y UTAUT. Finalmente, la evaluación del rendimiento de los estudiantes en los proyectos dependió completamente de la percepción de los profesores que los supervisaron.

- Validez externa: Se refiere a la medida en que es posible generalizar los hallazgos y en qué medida éstos son de interés para otras personas fuera del caso investigado. Pueden surgir problemas de validez externa dadas las características particulares de la evaluación: cantidad de estudiantes participantes, distancias geográficas de las universidades, diferencias culturales, así como el tamaño y complejidad de los proyectos desarrollados.

5. Conclusiones

A lo largo del desarrollo de esta tesis se ha expuesto la importancia del enfoque del GSD para la industria actual del desarrollo de software. Se abordaron específicamente los diferentes beneficios que se obtienen de la aplicación del enfoque en las empresas, dentro de los cuales se encuentran: el abaratamiento de costos, el acercamiento a mercados emergentes al contar con personal en la ubicación del cliente, la innovación que puede surgir de la unión de diferentes ideas y formas de trabajo, entre otros. No obstante, estos beneficios son difíciles de obtener en la práctica, puesto que el GSD presenta diversos problemas y retos que dificultan su implementación exitosa en las empresas, los cuales están relacionados directamente con el concepto de las tres distancias definidos por Ågerfalk et al. (2005) como:

- Distancia geográfica: Representa el esfuerzo requerido por una persona para visitar un sitio alejado de otro.
- Distancia temporal: Representa la deslocalización en tiempo que una persona experimenta para interactuar con otra.
- Distancia sociocultural: Representa la medida en la que un individuo comprende las costumbres y cultura de otro.

Con relación a lo anterior, en la literatura se reportan diferentes soluciones propuestas para disminuir los problemas y retos de la implementación del GSD en las empresas, de entre las cuales destaca la mejora en la enseñanza de este tópico en las universidades. Sin embargo, la educación del GSD es también un reto que debe abordarse, puesto que la enseñanza de la IS en general aún sigue un enfoque de clases magistrales, en donde los profesores son emisores de conceptos y los estudiantes tratan de retener la mayor cantidad de información posible, por lo cual este método no se ajusta adecuadamente a la enseñanza del GSD. Además, como se presentó en la sección 2.1.1 de esta tesis, la enseñanza del GSD en las universidades enfrenta diversos retos como: la distancia global entre universidades, los equipos de trabajo, los planes de estudio, los *stakeholders* y roles, la infraestructura técnica de cada universidad, los problemas relacionados con las personas, y los procesos de desarrollo. Es importante mencionar que la educación del GSD generalmente se da a nivel Posgrado y no se considera en la preparación de estudiantes de Licenciatura, como se pudo apreciar en los resultados del análisis sobre la enseñanza del GSD en las universidades, presentado en la sección 3.1.1 de este documento. Mejor aún, este estudio permitió evidenciar que en México no se considera al GSD como parte de la formación de los estudiantes universitarios, a pesar de que está siendo utilizado por la industria actual del software.

Considerando lo anterior, esta tesis abordó el problema de la enseñanza del GSD en las universidades y se enfocó en mejorar las habilidades suaves de los estudiantes con respecto al trabajo en equipo y a la comunicación en entornos globales. De esta manera, en el Capítulo 3 se presentó el diseño y definición de un marco de referencia que define los lineamientos para el desarrollo de cursos de GSD a nivel licenciatura. Al mismo tiempo, se desarrolló una plataforma computacional que lleva por nombre Cadxela GSD (o “descubre el GSD” en Zapoteco) que permite poner en práctica los lineamientos de dicho marco de referencia. Es importante mencionar que el marco de referencia implementa el método de enseñanza del ABP basado en la teoría constructivista para fomentar la enseñanza práctica del GSD.

Con el objetivo de evaluar la efectividad de la propuesta, se diseñó y condujo un caso de estudio que contó con la participación y colaboración de profesores y estudiantes de cinco universidades mexicanas, las cuales cumplieron con las características suficientes para establecer un curso de GSD, siguiendo los lineamientos del marco de referencia propuesto a través del uso de Cadxela GSD. Los resultados obtenidos permitieron validar la hipótesis planteada al inicio de la tesis, al mismo tiempo que evidenció los problemas y retos presentes en el GSD tanto para su enseñanza en las universidades como para su uso en un entorno de trabajo. Con respecto a la enseñanza del GSD en las universidades, se enfrentaron diversos problemas para organizar el curso, puesto que fue difícil encontrar universidades, profesores, y estudiantes que participaran en el caso de estudio, debido a que cada universidad tiene ya un ritmo de trabajo bien establecido y siguen los lineamientos de un plan de estudio. Por lo tanto, resultó difícil el establecer un momento y horario para trabajar en un curso externo que requería de un mayor esfuerzo. Así mismo, los profesores dedicaron tiempo extra en la ayuda y revisión de los estudiantes, por lo cual hubo profesores que estuvieron más involucrados que otros. Lo anterior pudo influenciar directamente en el trabajo de los equipos. También los estudiantes tuvieron que trabajar horas extra, ya que a la par tomaban cursos propios de su formación, por lo tanto, su trabajo se vio afectado por este hecho.

Por otro lado, los estudiantes enfrentaron los problemas presentes en el GSD, tales como:

- Problemas para comunicarse con sus compañeros.
- Problemas para trabajar en equipo.
- Problemas técnicos, en cuanto a los diferentes niveles de conocimiento de cada uno de sus compañeros.
- Problemas técnicos de infraestructura, ya que algunos estudiantes no contaban con un buen servicio de internet por lo que su desempeño y participación en la comunicación se vio directamente afectado.

Los resultados del caso de estudio proporcionaron evidencia de que la propuesta es exitosa en cuanto a mejorar las habilidades de comunicación y trabajo en equipo de los estudiantes a través de la ejecución de diferentes proyectos. Así mismo, los resultados obtenidos demuestran que en general los estudiantes estuvieron motivados y satisfechos con su trabajo y el enfoque de GSD establecido.

De esta manera, diferentes enfoques y herramientas, como los presentados en esta tesis, están enfocados en mejorar la enseñanza del GSD en las universidades. De manera similar, en la literatura se reportan diversos cursos de GSD entre universidades, pero en general son cursos a nivel de maestría y en universidades del extranjero, por lo cual, es importante ofrecer este tipo de alternativas a nivel licenciatura y que puedan ser llevados a cabo en México, ya que como se mencionó

anteriormente, este tema no es considerado en general por las universidades dentro de sus planes de estudio.

Finalmente, esta tesis permite establecer algunas líneas de trabajo futuro que pueden seguirse a partir de los resultados alcanzados:

- Diseñar otro caso de estudio con una muestra más grande de estudiantes y profesores, además de incluir a universidades del extranjero.
- Desarrollar una aplicación móvil que ayude a los estudiantes a comunicarse de mejor manera e implementar la comunicación informal que no se dio en este caso de estudio.
- Incluir en Cadxela GSD una herramienta que permita monitorear la construcción de los proyectos solicitados a los estudiantes.
- Incluir proyectos de mayor complejidad para que sean desarrollados por los estudiantes.

6. Anexo A.- Acrónimos

ABP	Aprendizaje basado en proyectos
ACM	Asociación para la maquinaria informática
BD	Base de datos
CEPPE	Comité para la evaluación de programas de pedagogía y educación
CF	Condiciones facilitadoras
CMU	Universidad de Carnegie Mellon
CONREDES	Consejo regional para el desarrollo de la educación y la sustentabilidad
CSCL	Aprendizaje colaborativo asistido por computadora
CWS	Esferas de trabajo colaborativo
EE	Expectativa de esfuerzo
EF	Factor de complejidad del entorno
ER	Expectativa de rendimiento
FTP	Protocolo de transferencia de archivos
FU	Facilidad de uso percibida
GSD	Desarrollo global de software
GUI	Interfaz gráfica de usuario
IEEE	Instituto de ingenieros eléctricos y electrónicos
IS	Ingeniería de software
IT	Tecnologías de la información
ITS	Sistemas inteligentes de tutoría
MUST	Universidad de Ciencia y Tecnología de Mongolia
NIT	Instituto Tecnológico de Nuremberg
PBL	Aprendizaje basado en problemas
PCA	Espacios de colaboración potenciales
PF	Factor de productividad

PyMES	Pequeñas y medianas empresas
RAD	Desarrollo rápido de aplicaciones
SI	Sistemas de información
SEEK	Guía de conocimiento sobre la educación de la ingeniería de software
SWEBOK	Cuerpo de conocimientos de la ingeniería de software
TAM	Modelo de aceptación de la tecnología
TCF	Factor de complejidad técnica
TUM	Universidad Técnica de Múnich
UAW	Peso de los actores sin ajustar
UCP	Puntos de caso de uso
UP	Utilidad percibida
UUCP	Puntos de casos de uso sin ajustar
UUCW	Pesos de casos de uso sin ajustar
UTAUT	Modelo de la teoría unificada de aceptación del uso de la tecnología
UTM	Universidad Tecnológica de la Mixteca
VPN	Redes virtuales privadas
WS	Esferas de trabajo

7. Bibliografía

Ågerfalk, P. J., Fitzgerald, B., Holmström O. H., y Conchúir, E. (2008). Benefits of global software development: The known and unknown. In Wang, Q., Pfahl, D., y Raffo D.M. (Eds.), *Making Globally Distributed Software Development a Success Story*, Berlin, Heidelberg: Springer-Verlag, pp. 1-9.

Ågerfalk, P. J., Fitzgerald, B., Holmström O. H., Lings, B., Lundell, B., y Conchúir, E. (2005). A framework for considering opportunities and threats in distributed software development. *Proc. of the International Workshop on Distributed Software Development*, Austrian Computer Society, pp. 47-61.

Ahmed, F., Capretz, L., Bouktif, S., y Campbell, P. (2012). Soft skills requirements in software development jobs: A cross-cultural empirical study. *Journal of Systems and Information Technology*, 14(1): 58-81.

Al-Qeisi, K. I. (2009). Analyzing the use of UTAUT model in explaining an online behaviour: Internet banking adoption (Doctoral dissertation, Brunel Business School, Brunel University, Uxbridge, UK).

Ardis, M. A., Chenoweth, S. V., y Young, F. H. (2008). The “soft” topics in software engineering education. *Proc. of the 38th Annual Frontiers in Education Conference (FIE 2008)*, IEEE Computer Society, pp. F3H-1-F3h-6.

Ardis, M. A., Budgen, D., Hislop, G. W., Offutt, J., Sebern, M. J., & Visser, W. (2015). SE 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. *IEEE Computer*, 48(11), 106-109.

Bass, J. M., McDermott, R., y Lalchandani, J. T. (2015). Virtual teams and employability in global software engineering education. *Proc. of the 10th International Conference on Global Software Engineering (ICGSE)*, IEEE Computer Society, pp. 115-124.

Beecham, S., Clear, T., y Noll, J. (2017). Do we teach the right thing?: A comparison of global software engineering education and practice. *Proc. of the 12th International Conference on Global Software Engineering*, IEEE Computer Society, pp. 11-20.

Bhatti, M. W. y Ahsan, A. (2016). Global software development: An exploratory study of challenges of globalization, HRM practices and process improvement. *Review of Managerial Science*, 10(4): 649-682.

Bosnić, I., Žagar, M., y Crnković, I. (2014). Technology in real-life teaching of distributed software development. *EUNIS Journal of Higher Education*, (1): 89-97.

Bourque, P. y Fairley, R. E. (2014). Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0. Los Alamitos, CA: IEEE Computer Society Press.

Braun, A., Dutoit, A. H., Harrer, A. G., y Brugge, B. (2002). IBistro: A learning environment for knowledge construction in distributed software engineering courses. *Proc. of the 9th Asia-Pacific Software Engineering Conference*, IEEE Computer Society, pp. 197-203.

Buchanan, S. M. C. (2018). The lived experience of middle school students engaged in student-driven inquiry: A phenomenological study (Doctoral dissertation, Queensland University of Technology. Brisbane City, Australia).

Cabero, J. (2006). Bases pedagógicas del e-learning. *Revista de Universidad y Sociedad del Conocimiento*, 3(1): 1-10.

Carmel, E. y Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *IEEE software*, 18(2): 22-29.

Casey, V. y Richardson, I. (2008). The impact of fear on the operation of virtual teams. *Proc. of the IEEE International Conference on Global Software Engineering (ICGSE 2008)*, IEEE Computer Society, pp. 163-172.

Cho, J. (2007). Globalization and global software development. *Issues in Information Systems*, 8(2): 287-290.

Clear, T., Beecham, S., Barr, J., Daniels, M., McDermott, R., Oudshoorn, M., Savickaite, A., y Noll, J. (2015). Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review. *Proc. of the 20th Annual Conference on Innovation and Technology in Computer Science Education on Working Group Reports*, ACM Publisher, pp. 1-39.

Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., García-Peñalvo, F. J., y Tovar, E. (2014). Project managers in global software development teams: a study of the effects on productivity and performance. *Software Quality Journal*, 22(1): 3-19.

Conchúir, E. (2010). Global software development: A multiple-case study of the realisation of the benefits (Doctoral Thesis, University of Limerick. Limerick, Ireland).

Cohn, M. (2004). *User stories applied for agile software development*. Boston, MA: Addison-Wesley.

Cohn, M. (2005). *Agile estimating and planning*. Upper Saddle River, NJ: Pearson Education.

Cohn, M. (2010). *Succeeding with agile: software development using Scrum*. Upper Saddle River, NJ: Pearson Education.

Consejo Regional para el Desarrollo de la Educación y la Sustentabilidad. (2017). *Demanda de talento desde la perspectiva de la industria*. Recuperado de <http://www.uacj.mx/DGVI/sv/Documents/Demanda%20de%20Talento.pdf>

Crnković, I., Bosnić, I., y Žagar, M. (2012). Ten tips to succeed in global software engineering education. *Proc. of the 34th International Conference on Software Engineering (ICSE)*, IEEE Computer Society, pp. 1225-1234.

Damian, D. y Moitra, D. (2006). Guest editors' introduction: Global software development: How far have we come? *IEEE Software*, 23(5): 17-19.

Ebert, C. y De Neve, P. (2001). Surviving global software development. *IEEE software*, 18(2): 62-69.

Ende, M., Lämmermann, R., Brockmann, P., y Ayurzana, G. (2013). A virtual, global classroom to teach global software engineering: A Mongolian-German team-teaching project. *Proc. of the Second International Conference on e-Learning and e-Technologies in Education (ICEEE)*, IEEE Computer Society, pp. 229-233.

Fagerholm, F., Oza, N., y Münch, J. (2013). A platform for teaching applied distributed software development: The ongoing journey of the Helsinki software factory. *Proc. of the 3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD)*, IEEE Computer Society, pp. 1-5.

Fortaleza, L. L., Conte, T., Marczak, S., y Prikładnicki, R. (2012). Towards a GSE international teaching network: Mapping global software engineering courses. *Proc. of the Second International Workshop on Collaborative Teaching of Globally Distributed Software Development*, IEEE Computer Society, pp. 1-5.

Ghanbari, H., Similä, J., y Markkula, J. (2015). Utilizing online serious games to facilitate distributed requirements elicitation. *Journal of Systems and Software*, 109: 32-49.

Gotel, O., Kulkarni, V., Scharff, C., y Neak, L. (2008). Working across borders: Overcoming culturally-based technology challenges in student global software development. *Proc. of the IEEE 21st Conference on Software Engineering Education and Training (CSEET'08)*, IEEE Computer Society, pp. 33-40.

Göl, Ö. y Nafalski, A. (2007). Collaborative learning in engineering education. *Global Journal of Engineering Education*, 11(2): 173-180.

Hadjerrouit, S. (2005). Constructivism as guiding philosophy for software engineering education. *ACM SIGCSE Bulletin*, 37(4): 45-49.

Herbsleb, J. D. y Moitra, D. (2001). Global software development. *IEEE Software*, 18(2): 16-20.

Holmström, H., Fitzgerald, B., Ågerfalk, P. J., y Conchúir, E. (2006). Agile practices reduce distance in global software development. *Information Systems Management*, 23(3): 7-18.

Holtkamp, P., Lau, I., y Pawlowski, J. M. (2015). How software development competences change in global settings—an explorative study. *Journal of Software: Evolution and Process*, 27(1): 50-72.

Jiménez, M., Piattini, M., y Vizcaíno, A. (2009). Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering*, (3): 1-15.

Juristo, N. & Moreno, A. (2010). *Basics of software engineering experimentation*. Dordrecht, the Netherlands: Kluwer Academic Press.

Kolbe, T. y Jorgenson, S. (2018). Meeting instructional standards for middle-level science: Which teachers are most prepared? *The Elementary School Journal*, 118(4): 1-29.

Krajcik, J. S. y Blumenfeld, P. C. (2006). Project-based learning. In Sawyer, R. K. (Ed.), *The Cambridge Handbook of the Learning Sciences*, New York, NY: Cambridge, pp. 317-34.

Lappalainen, J., Tripathi, N., y Similä, J. (2016). Teaching a global software development course: Student experiences using onsite exercise simulation. *Proc. of the 38th International Conference on Software Engineering Companion*, ACM Publisher, pp. 440-450.

Lee, J. N., Huynh, M. Q., Chi-Wai, K. R., y Pi, S. M. (2000). The evolution of outsourcing research: What is the next issue? *Proc. of the 33rd Annual Hawaii International Conference on System Sciences*, IEEE Computer Science, pp. 1-10.

Li, Y., Krusche, S., Lescher, C., y Bruegge, B. (2016). Teaching global software engineering by simulating a global project in the classroom. *Proc. of the 47th ACM Technical Symposium on Computing Science Education*, ACM Publisher, pp. 187-192.

Matthes, F., Neubert, C., Schulz, C., Lescher, C., Contreras, J., Laurini, R., Rumpler, B., Sol, D., y Warendorf, K. (2011). Teaching global software engineering and international project

management. *Proc. of the 3rd International Conference on Computer Supported Education*, SciTePress Digital Library, pp. 5-15.

Matturro, G., Raschetti, F., y Fontán, C. (2015). Soft skills in software development teams: A survey of the points of view of team leaders and team members. *Proc. of the 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, IEEE Computer Society, pp. 101-104.

Mills, H. D., O'Neill, D., Linger, R. C., Dyer, M., y Quinnan, R. E. (1980). The management of software engineering. *IBM Systems Journal*, 24(2): 414-477.

Monasor, M. J. (2014). A framework for interaction training in Global Software Development (Tesis Doctoral, Universidad de Castilla - La Mancha. Ciudad Real, España).

Monasor, M. J., Vizcaíno, A., y Piattini, M. (2011). Docencia en desarrollo global de software: Una revisión sistemática. *Jornadas de Enseñanza Universitaria de la Informática*, Universidad de Sevilla, pp. 221-228.

Monasor, M. J., Vizcaino, A., Piattini, M., y Caballero, I. (2010). Preparing students and engineers for global software development: A systematic review. *Proc. of the 5th IEEE International Conference on Global Software Engineering (ICGSE)*, IEEE Computer Society, pp. 177-186.

Nestel, D., & Tierney, T. (2007). Role-play for medical students learning about communication: Guidelines for maximising benefits. *BMC medical education*, 7(1): 1-9.

Nii, H. P. (1986). Blackboard application systems, blackboard systems and a knowledge engineering perspective. *AI Magazine*, 7(3): 82-107.

Noll, J., Beecham, S., y Richardson, I. (2010). Global software development and collaboration: Barriers and solutions. *ACM Inroads*, 1(3): 66-78.

Noll, J., Butterfield, A., Farrell, K., Mason, T., McGuire, M., y McKinley, R. (2014). GSD Sim: A global software development game. *Proc. of the IEEE International Conference on Global Software Engineering Workshops (ICGSEW)*, IEEE Computer Society, pp. 15-20.

Ochodek, M., Nawrocki, J., y Kwarciak, K. (2011). Simplifying effort estimation based on Use Case Points. *Information and Software Technology*, 53(3): 200-213.

O'Neill, G. (2015). *Curriculum design in higher education: Theory to practice*. 1st Edition. Dublin, Irlanda: University College Dublin.

Pacheco, C. y Garcia, I. (2012). A systematic literature review of stakeholder identification methods in requirements elicitation. *Journal of Systems and Software*, 85: 2171-2181.

Palacio, R. R., Vizcaíno, A., Morán, A. L., y González, V. M. (2011). Tool to facilitate appropriate interaction in global software development. *IET software*, 5(2): 157-171.

Paasivaara, M., Lassenius, C., Damian, D., Rätty, P., y Schröter, A. (2013). Teaching students global software engineering skills using distributed Scrum. *Proc. of the 35th International Conference on Software Engineering (ICSE)*, IEEE Computer Society, pp. 1128-1137.

Ribaud, V. y Saliou, P. (2015). Introducing problem-based learning in a joint masters degree: Offshoring information technologies. In Sobh, T., Elleithy, K. (Eds.), *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*. Springer, Cham, pp. 311-319.

Richardson, I., Moore, S., Paulish, D., Casey, V., y Zage, D. (2007). Globalizing software development in the local classroom. *Proc. of the 20th Conference on Software Engineering Education & Training (CSEET'07)*, IEEE Computer Society, pp. 64-71.

- Rojas, R. A. (1998). La metodología del cuestionario. *La sociología en sus escenarios*, (1): 1-15.
- Runeson, P. y Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2): 131-164.
- Sagi-Vela, L. (2004). Gestión por competencias: El reto compartido del crecimiento personal y de la organización. Madrid, España: Esic Editorial.
- Saldaña-Ramos, J., Sanz-Esteban, A., García, J., y Amescua, A. (2014). Skills and abilities for working in a global software development team: A competence model. *Journal of Software: Evolution and Process*, 26(3): 329-338.
- Schank, R. C., Berman, T. R., y Macpherson, K. A. (1999). *Learning by doing. Instructional-design theories and models*. Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Sheehan, J. (1986). Curriculum models: Product versus process. *Journal of Advanced Nursing*, 11(6): 671-678.
- Šmite, D., Wohlin, C., Gorschek, T., y Feldt, R. (2010). Empirical evidence in global software engineering: A systematic review. *Empirical Software Engineering*, 15(1): 91-118.
- Stein, D. (1998). *Situated learning in adult education*. Columbus, OH: ERIC Clearinghouse on Adult Career and Vocational Education.
- Valencia, D., Vizcaíno, A., Garcia-Mundo, L., Piattini, M., y Soto, J. P. (2016). GSDgame: A serious game for the acquisition of the competencies needed in GSD. *Proc. of the IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)*, IEEE Computer Society, pp. 19-24.
- Vardi, M. Y. (2010). Globalization and offshoring of software. Revisited. *Communications of the ACM*, 53(5): 5-5.
- Vizcaíno, A., García, F., y Piattini, M. (2015). Visión general del desarrollo global de software. *International Journal of Information Systems and Software Engineering for Big Companies*, 1(1): 8-22.
- Vizcaíno, A., García, F., Piattini, M., y Beecham, S. (2016). A validated ontology for global software development. *Computer Standards & Interfaces*, 46: 66-78.
- Weisberg, H. F. (2009). *The total survey error approach*. Chicago, IL: University of Chicago Press.