

**Universidad ORT Uruguay
Facultad de Ingeniería**

Aprendizaje activo en ingeniería de software

Entregado como requisito para la obtención del título de Ingeniero en Sistemas

Juan Ignacio Irabedra - 212375

Tutor: Gastón Mousqués

Tutor: Martín Solari

2024

Declaración de autoría

Yo, Juan Ignacio Irabedra, declaro que el trabajo que se presenta en esa obra es de mi propia mano. Puedo asegurar que:

- La obra fue producida en su totalidad mientras realizaba el proyecto;
- Cuando he consultado el trabajo publicado por otros, lo he atribuido con claridad;
- Cuando he citado obras de otros, he indicado las fuentes. Con excepción de estas citas, la obra es enteramente mía;
- En la obra, he acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, he explicado claramente qué fue contribuido por otros, y qué fue contribuido por mí;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

A handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke extending to the right, enclosed within a faint oval outline.

Juan Ignacio Irabedra

8 de enero de 2024

Agradecimientos

A mis tutores, Martín y Gastón, quienes me apoyaron y guiaron, siempre contagiando su pasión y vocación por la ingeniería de software y su enseñanza. Ha sido un privilegio que me acompañen en este proyecto.

A mis compañeros de la Cátedra de Ingeniería de Software, que me acompañan en el crecimiento día a día.

A mis compañeros de Montevideo Labs, y en particular a Nicolás y Victoria por apoyarme incondicionalmente.

A mis amigos, por su incesante aliento.

A mis estudiantes y docentes, por inspirarme clase a clase.

Abstract

Contexto: La ingeniería de software es una disciplina intrínsecamente aplicada y la industria demanda cada vez más profesionales con experiencia en la aplicación de conocimientos y uso de herramientas tecnológicas en contextos de equipos interdisciplinarios. En los últimos años se han desarrollado distintas experiencias de aprendizaje activo en educación superior para mejorar la preparación de los profesionales en el área. La Cátedra de Ingeniería de Software de Universidad ORT Uruguay también lleva adelante actividades instruccionales que buscan preparar apropiadamente a los futuros profesionales del software.

Objetivo: Los objetivos generales de este proyecto de investigación son encontrar e identificar las estrategias de aprendizaje activo en ingeniería de software (1) así como también experiencias de aplicación en educación superior asociadas a tales estrategias (2), y presentar y reflexionar acerca de las actividades de aprendizaje activo en la Cátedra, con un foco en la aplicación y evolución del uso de Team-Based Learning (3).

Método: Para los objetivos (1) y (2) se condujo un estudio secundario con método de Rapid Review. Una Rapid Review es un tipo de estudio secundario que acelera un proceso de un mapeo sistemático de literatura. Para el objetivo (1) también se consultaron sitios web de centros especializados en educación de universidades destacadas en el mundo. Para el objetivo (3) se consultaron documentos oficiales de la Universidad ORT Uruguay y se realizaron 6 entrevistas semi-estructuradas a docentes.

Resultados: Para el objetivo (1) se identificaron 45 estrategias de aprendizaje activo. 18 de ellas vienen de estudios realizados entre 2018 y 2023 alcanzados mediante la Rapid Review. Para el objetivo (2) se identificaron 49 estudios, 41 de ellos primarios. Juegos y Project-Based Learning son las estrategias que más aparecen. Existen experiencias que combinan diferentes estrategias de aprendizaje activo. Para el objetivo (3) se reporta el desafío de implementar estrategias de aprendizaje activo en entornos de enseñanza híbridos o remotos. Estas estrategias deben ser combinadas con actividades tradicionales. El cuerpo docente destaca positivamente el impacto de Team-Based Learning y aprendizaje activo en cursos de ingeniería de software.

Conclusiones: Existen estrategias de aprendizaje activo, en auge en educación superior en ingeniería de software que fomentan el desarrollo de conocimientos y competencias. El diseño instruccional puede enriquecerse de tomar estrategias de aprendizaje activo y combinarlas con estrategias de aprendizaje tradicional para generar experiencias en el aula más cercanas a la práctica profesional. La Cátedra de Ingeniería de Software de la Facultad de Ingeniería de Universidad ORT Uruguay ha adaptado prácticas instruccionales, y en particular Team-Based Learning, en pos de mejorar la experiencia tanto docente como estudiante.

Palabras clave

Aprendizaje activo, Ingeniería de Software, Educación, Constructivismo

Indice

1.	Introducción.....	9
2.	Marco teórico.....	12
2.1.	Ingeniería de software	12
2.2.	Constructivismo.....	12
2.3.	Pensamiento de orden superior.....	13
2.4.	Aprendizaje significativo.....	14
2.5.	Competencias	15
2.6.	Habilidades blandas	16
2.7.	Aprendizaje tradicional.....	16
2.8.	Aprendizaje activo	17
2.9.	Team-Based Learning.....	18
3.	Trabajos relacionados	21
4.	Estrategias y experiencias de aprendizaje activo en ingeniería de software	24
4.1.	Introducción.....	24
4.2.	Preguntas de investigación (PI)	25
4.3.	Método.....	25
4.3.1.	Búsqueda	25
4.3.2.	Extracción de datos.....	27
4.4.	Resultados.....	28
4.4.1.	PI1: Estrategias de aprendizaje activo en ingeniería de software.....	28
4.4.2.	PI2: Experiencias de aplicación de las estrategias de aprendizaje activo.....	29
4.5.	Discusión de estrategias de aprendizaje activo.....	30
4.5.1.	Introducción a las estrategias de aprendizaje activo.....	31

4.5.2.	Marcos pedagógicos y técnicas de aprendizaje activo	38
	Estrategias generales y de aplicación de actividades de desarrollo de software	40
4.5.3.	Estrategias grupales e individuales.....	42
4.5.4.	Combinación de estrategias de aprendizaje activo	42
4.6.	Amenazas a la validez	43
5.	Aprendizaje activo en ingeniería de software en Universidad ORT Uruguay	44
5.1.	Introducción.....	44
5.1.	Contexto y definiciones previas	45
5.2.	Pregunta de investigación.....	50
5.3.	Método.....	51
5.4.	Resultados.....	52
5.4.1.	Evaluaciones y puntajes.....	52
5.4.2.	Evolución de la aplicación de TBL	54
5.4.3.	Entrevistas con docentes.....	57
5.5.	Discusión de actividades y evaluaciones en la Cátedra de Ingeniería de Software en Universidad ORT Uruguay	59
5.5.1.	Actividades y evaluaciones en la Cátedra	59
5.5.2.	Despliegue de Team-Based Learning.....	61
5.5.3.	Reflexiones de aprendizaje activo en ingeniería de software.....	63
5.6.	Amenazas a la validez	64
6.	Conclusiones.....	66
6.1.	Líneas de trabajo a futuro	67
7.	Referencias bibliográficas	68
	ANEXO 1	79
	ANEXO 2	80

Índice de ilustraciones

Ilustración 1: secuencia de actividades de TBL	19
Ilustración 2: clasificación de estrategias según marco pedagógico o técnica	39
Ilustración 3: clasificación de estrategias según tipo de actividad	41
Ilustración 4: materias de la Cátedra y sus previaturas	46
Ilustración 5: mecanismo de evaluación de DA2, recuperado de sitio de Bedelías	48
Ilustración 6: actividades para pregunta de investigación 3	51
Ilustración 7: distribución de puntaje total en la Cátedra	53
Ilustración 8: distribución de puntaje clasificado por materia y total relativo en la Cátedra.....	54
Ilustración 9: hitos relevantes en la evolución de ISA e ISA1	55
Ilustración 10: hitos relevantes en la evolución de AR	55
Ilustración 11: hitos relevantes en la evolución de ISA2	56

1. Introducción

La ingeniería de software (IS) tiene como objetivo el desarrollo, la operación y la evolución de productos digitales en equipos interdisciplinarios bajo restricciones de calidad, tiempo y costos. Estas actividades se realizan con un enfoque sistémico, disciplinado y cuantificable [1].

La manera de llevar adelante las actividades de IS ha cambiado con el tiempo, y distintos paradigmas han primado en la industria como lo es el caso de la Ingeniería de Software Ágil (ISA) [2]. La ISA ha divergido hacia una disciplina con claras diferencias respecto a la ingeniería tradicional: ha puesto el foco en la respuesta al cambio ante la planificación exhaustiva, las interacciones interpersonales ante la negociación por contratos y la entrega frecuente de productos digitales en funcionamiento ante extensa documentación.

En este contexto, una de las tendencias más en auge en la industria es el interés por las llamadas habilidades blandas [3]. Este conjunto de habilidades, como la empatía, la buena disposición y la cooperación ya no son un factor deseable en los profesionales, sino un requisito incluso presente en la descripción de ofertas laborales [4].

Esta realidad que engloba a la ISA deja en evidencia que es una disciplina en constante cambio y con características que la diferencian respecto a otros tipos de ingenierías. Por tanto, requiere que los futuros profesionales del área tengan una formación adaptada a esta realidad. Los desafíos propios de la IS, que ya fueron mencionados, determinan que esta disciplina no puede ser enseñada exclusivamente mediante actividades realizadas tradicionalmente en un salón de clase [5].

Algunos estudios recientes muestran como la enseñanza de la IS no se ajusta a las necesidades de la industria [6][7]. Existen importantes brechas entre la formación de profesionales en ingeniería de software y las expectativas de la industria. Algunas de ellas residen en las habilidades blandas, en los procesos de gestión de IS y en la gestión de la configuración [7]; es decir desafíos inherentes a la práctica profesional que, nuevamente, son difícilmente alcanzables en un salón de clase.

Es relevante entonces indagar qué estrategias existen para enseñar conocimientos y habilidades de IS. Una de las estrategias generales con más relevancia en las últimas décadas es aprender haciendo [8]. Aprender haciendo es una estrategia o grupo de estrategias para acercar a los futuros profesionales a las actividades que formarán parte de su práctica profesional. Cabe preguntar, entonces, cómo se puede vehicular esta relación aprendizaje-enseñanza de modo tal que ocurra en un contexto de aprender haciendo: ¿cuáles son o qué tipo de actividades o dinámicas permiten aprender haciendo?

El aprendizaje activo (AA) agrupa estrategias de aprendizaje que favorecen el aprender haciendo. El AA ha polarizado opiniones en el ámbito educativo e invita a pensar si existe algún tipo de aprendizaje no activo. Además, los académicos en el área de ingeniería no siempre indagan en las estrategias de AA o las diferencias que tienen con estilos de aprendizaje tradicionales. Tampoco están inclinados a buscar respuestas en la literatura sobre educación. Sin embargo, el AA ha demostrado permitir alcanzar mejores resultados y experiencias positivas para los estudiantes de ingeniería [9].

Este estudio recoge de estudios primarios y secundarios qué estrategias existen y son recientemente aplicadas para implementar el AA en educación superior en IS. El principal método elegido es de Rapid Review (RR) [10]. Este método permite navegar el espacio de búsqueda de una revisión sistemática de literatura de manera acotada, veloz y menos costosa. El método de RR sacrifica precisión y alcance en comparación a una revisión sistemática de literatura. Sin embargo, el método de RR favorece particularmente a este estudio ya que es un estudio dual. No solo se exploran las estrategias de AA implementadas en los años más recientes en educación superior, sino que también se vuelcan los resultados de búsqueda de estrategias en la presentación y reflexión sobre AA en la Cátedra de Ingeniería de Software de la Universidad ORT Uruguay.

El presente estudio busca mirar hacia el mundo y conocer el estado del AA en IS, para nutrir la toma de decisiones instruccionales en nuestra Universidad y para presentar parte de la experiencia de aplicación de estrategias de AA de la Cátedra. Para esto se utiliza un enfoque de búsqueda mixto: se contemplan los resultados de una búsqueda en

la literatura académica, así como también se busca en centros de apoyo a la enseñanza universitaria de algunas universidades.

La segunda línea de investigación que introduce este estudio involucra reflexionar acerca de las formas de enseñar IS en la Cátedra. En las distintas materias curriculares de la Cátedra se han implementado varias estrategias de aprendizaje, que han evolucionado de acuerdo con necesidades de la industria y la academia. Gran parte del conocimiento relacionado a esta evolución y planteo de las diferentes actividades vive de manera tácita en la memoria de miembros de la Cátedra, y es de interés documentarlo y sistematizarlo.

Este estudio, mediante entrevistas semi-estructuradas así como recolección de información y documentos de los cursos, reúne hitos importantes en el camino recorrido en la incorporación de distintas actividades de enseñanza-aprendizaje. La Cátedra lleva adelante diversas estrategias de AA como clases invertidas, ejercicios de desarrollo de software, mini-proyectos y trabajos en equipo. Este estudio además pone bajo la lupa la experiencia de aplicación de una estrategia de AA particular: Team-Based Learning (TBL).

La Cátedra tiene experiencia aplicando TBL desde el año 2015 con muchos ajustes que han atendido las pautas específicas de la Universidad sobre las evaluaciones y la evolución disciplinar de las materias. TBL es un método instruccional que pone el centro en equipos de trabajo, conjuga la responsabilidad colectiva e individual y promueve el AA [11].

Diseñar las actividades instruccionales y ajustarlas no es tarea sencilla; el equipo docente de IS intenta crecer manteniendo el compromiso de brindar una buena experiencia tanto docente como estudiante en el diseño e implementación de estas actividades. Por esta razón el último cometido de este estudio es recolectar y sintetizar el conocimiento acerca de las actividades realizadas anteriormente para servir como lecciones aprendidas al moverse hacia adelante.

2. Marco teórico

A continuación se presentan conceptos clave sobre los que se basa esta investigación.

2.1. Ingeniería de software

Existen definiciones técnicas de IS, como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la IS [1]. Esta definición es válida, pero también es algo rígida y normalizada; sirve como estándar, pero es más costoso intentar reflejar la esencia de la IS en el mundo actual.

Otras definiciones contemplan el contexto en el cual viven los proyectos y productos de software. Esta manera de entender la IS tiene en cuenta no solo los procesos técnicos de desarrollo, sino también los procesos de gestión, la naturaleza de las organizaciones que llevan adelante el desarrollo de software y las personas involucradas en el proceso de desarrollo de software [12]. Esta visión de IS plantea que no hay consenso universal respecto a los métodos y técnicas de IS, sino que existe un grupo de métodos y técnicas heterogéneo. Esta diversidad es la que los ingenieros de software deben navegar a los efectos de seleccionar y llevar adelante las mejores prácticas para el contexto en el cual viven los proyectos y productos de software.

Estas maneras de entender la IS conducen hacia tener en cuenta los procesos técnicos sin perder de vista la realidad dinámica y social en la cual vive el software. Por eso se entiende que la IS reúne actividades como el desarrollo, la operación, la validación y la evolución de productos digitales en equipos multidisciplinarios de manera sistemática bajo restricciones de calidad, tiempo y costos.

2.2. Constructivismo

El constructivismo es una perspectiva psicológica y filosófica que propone al aprendizaje como un proceso construido, en gran parte, por las personas [13]. La concepción constructiva del aprendizaje se opone a otros abordajes del aprendizaje que se centran en la influencia del entorno sobre el aprendiz o en la respuesta visible del aprendiz respecto al aprendizaje. Un elemento central en el aprendizaje constructivista

es el conocimiento previo que posee el sujeto que aprende. Este conocimiento previo se estructura mediante esquemas.

La información nueva obtenida por el aprendiz es ajustada para ser congruente con la estructura cognoscitiva preexistente. Este proceso se conoce como asimilación. Por otro lado, para poder darle sentido a la nueva información, la estructura cognoscitiva debe ajustarse para ser congruente con la realidad observada; este proceso se conoce como acomodación. Este desequilibrio tiene lugar cuando existen conflictos entre el nuevo conocimiento y la estructura cognoscitiva existente [14].

El abordaje constructivista del aprendizaje es relevante porque provee bases para planificar y adaptar las prácticas pedagógicas que pueden provocar cambios en el conocimiento a largo plazo y ponen el centro en el estudiante [15]. Del mismo modo, explica por qué no todas las personas aprenden del mismo modo y enfatiza la importancia del conocimiento previamente adquirido [16].

2.3. Pensamiento de orden superior

El pensamiento de orden superior agrupa los procesos cognitivos más complejos presentados por Bloom [17] en su taxonomía revisada. Los procesos de orden más bajo consisten en recordar, comprender y aplicar. Los procesos de alto orden consisten en analizar, evaluar y crear.

La taxonomía de Bloom y el pensamiento de orden superior son relevantes en el contexto de aprendizaje en ingeniería dado que este marco conceptual es ampliamente usado para el diseño curricular, brinda un lenguaje común, sencillo y preciso para comunicar objetivos de aprendizaje [18] y permite delimitar la amplitud y profundidad con la que se presentan unidades temáticas en un curso [19]. La naturaleza aplicada de la ingeniería propone desafíos profesionales que requieren tanto actividades de pensamiento de orden inferior como de orden superior. Las actividades de pensamiento de orden inferior son sencillamente alcanzadas por métodos instruccionales de tipo tradicional. En cambio, las actividades de pensamiento de orden superior requieren otro tipo de estrategias para ser logradas.

2.4. Aprendizaje significativo

Ausubel introdujo la noción de aprendizaje significativo. Se dice que el aprendizaje es significativo cuando exitosamente se relaciona con conocimiento anteriormente obtenido [20]. El aprendizaje es significativo cuando el nuevo conocimiento es efectivamente asimilado [21], es decir relacionado con las estructuras cognoscitivas existentes.

El aprendizaje significativo es profundo; la nueva información subsume en información previamente adquirida, habitualmente más general. Al integrarse con las estructuras cognoscitivas existentes, la nueva información estará disponible a largo plazo y podrá ser aplicada, en lugar de solamente recordada. El proceso mediante el cual el conocimiento nuevo subsumido en conocimiento más general puede ser diferenciado a pesar de las características comunes se conoce como conciliación integradora [22].

Si bien la nueva información puede subsumir en conocimientos previamente integrados en estructuras cognoscitivas existentes, también existen otras variables que pueden determinar el éxito de la asimilación, como el material instruccional, estilo cognitivo, habilidad intelectual, motivación, variables sociales y afectivas [23].

El aprendizaje significativo es trascendental en la formación a largo plazo de un individuo. Sin embargo, incluso el aprendizaje significativo puede dejar de ser efectivo a largo plazo. Ausubel habla de un proceso que puede explicar el olvido del aprendizaje significativo: la subsunción obliteradora [24]. Este tipo de subsunción se opone a la conciliación integradora pues se pierde la capacidad de pasar de la idea ancla general a la idea particular, y por tanto se olvida.

Otro elemento capital en el aprendizaje significativo de Ausubel son los organizadores previos [25]. Los organizadores previos son materiales provistos antes de presentar un tema que sirven para activar conocimientos previos y relacionarlos con conceptos nuevos. La abstracción, amplitud y generalidad de estos materiales deben ser mayor a los del tema a presentar. Este tipo de material es relevante para el aprendizaje significativo.

El aprendizaje significativo es un elemento central en muchas definiciones de aprendizaje activo.

2.5. Competencias

La idea de competencia aparece cada vez más frecuentemente en la jerga de educación. Una definición de competencia es la habilidad de integrar y aplicar conocimientos, habilidades y factores psicosociales (tales como creencias, valores y motivaciones) apropiados para un contexto dado para desempeñarse exitosa y consistentemente en un contexto determinado [26]. Este elemento de saber leer contextos y actuar en consecuencia es determinante en la concepción de competencias [27].

Ideas como conocimiento, habilidades, actitudes y valores aparecen en otras definiciones [28], también contextualizándolas en la vida profesional e incluso la sociedad. Lo que diferencia a las competencias de la simple combinación de estas ideas es la capacidad de integrarlas y activarlas en contextos auténticos [29]. Incluso pueden aparecer factores más al detalle en cuanto a qué hace a una competencia, como procedimientos, reglas y estándares.

Las competencias son un concepto relevante porque mediante ellas, el diseño instruccional no solo persigue conocimientos o habilidades aisladas, sino que persigue también objetivos interdisciplinarios y contextualizados. Son particularmente importantes en el contexto del aprendizaje activo en ingeniería porque permiten que los aprendizajes trasciendan el aula y formen profesionales preparados para las situaciones que la vida profesional en sociedad les ofrece.

Se habla también de meta-competencias. Las meta-competencias son un tipo particular de competencias que no se limitan a dominios particulares de aplicación [30]. Son habilidades de alto orden que determinan la capacidad de aprender, adaptarse, anticiparse y crear [31]. El desarrollo de meta-competencias es crítico para el desarrollo académico y profesional, especialmente en el área de software ya que los ecosistemas tecnológicos y de buenas prácticas pivotean de manera frecuente.

2.6. Habilidades blandas

La frecuencia con la que se puede encontrar la noción de habilidades blandas es cada vez más alta. Sin embargo, es una noción algo turbia y difícil de manejar [32]. Es difícil encontrar una definición universal de habilidades blandas, ya que agrupa una combinación de factores emocionales, comportamentales y cognitivos, además de estar recíprocamente ligadas a las habilidades duras o técnicas [33].

Existen cuatro componentes comunes a varias definiciones de habilidades blandas: habilidades, actitudes, hábitos y rasgos de personalidad [3]. Con estos elementos presentes, se pueden entender las habilidades blandas como una combinación de habilidades, actitudes, hábitos y rasgos de personalidad que permiten a las personas desempeñarse mejor en el ámbito profesional y complementan las habilidades técnicas e influyen en sus interacciones interpersonales.

Las habilidades blandas son de particular interés en el contexto de IS ya que desde el año 2000 constituyen un requerimiento para los profesionales en la industria: algunas habilidades blandas son demandadas en ofertas laborales de manera afín a las habilidades duras o técnicas [4]. Las habilidades blandas son necesarias para enfrentar situaciones propias de la vida profesional, y es ventajoso exponer a los estudiantes a situaciones reales de la vida profesional para desarrollar habilidades blandas [34].

Algunos autores también usan habilidades blandas como concepto íntimamente ligado a algunas meta-competencias [30] como el pensamiento crítico y la capacidad de autoevaluación [35].

2.7. Aprendizaje tradicional

En este estudio se entiende como aprendizaje tradicional todos aquellos métodos instruccionales no activos. La principal actividad llevada adelante en entornos de aprendizaje tradicionales es la lección o clase magistral [36][37][38].

Los métodos instruccionales tradicionales exhiben características positivas como valorar el conocimiento per se, fomentar el ejercicio del pensamiento de orden inferior y ser parte de la cultura educativa actual. No obstante, también presentan desventajas: existen

situaciones en las cuales los estudiantes de menor rendimiento obtienen peores resultados con estos métodos instruccionales [36][38] o los estudiantes no tienen la posibilidad de practicar el pensamiento divergente [37].

2.8. Aprendizaje activo

Los estudiantes no aprenden mucho sentándose en clases escuchando a profesores o memorizando lecciones diseñadas para ser repetidas [39]. Sin embargo, existen académicos que defienden la idea de que el aprendizaje es inherentemente activo, y que incluso sentarse a escuchar una lección implica involucramiento activo por parte de los estudiantes [40].

Los estudiantes deben hacer más que solo escuchar para aprender [39]. Deben al menos leer, discutir, resolver problemas; o lo que es mejor, realizar actividades de pensamiento de orden superior, como análisis, síntesis, evaluación o creación [40].

El AA se define como cualquier método instruccional que involucra a los estudiantes en el proceso de aprendizaje mediante la realización de actividades de aprendizaje significativo [9] y pensar sobre aquello que están haciendo.

El AA ha demostrado ser efectivo en ingeniería [9], y las estrategias de AA pueden coexistir y complementar estrategias tradicionales [41]. El AA permite salir del aprendizaje tradicional de ingeniería de tipo deductivo (comenzar por la regla general y volcarla a casos particulares) y dar lugar a aprendizaje de tipo inductivo (comenzar con problemas del mundo real o situaciones particulares para construir la regla general) [42].

El AA contempla actividades para las cuales no existe una única solución y cada estudiante puede realizar actividades que lo lleven a una respuesta diferente, siguiendo los principios del aprendizaje constructivista. Las actividades significativas que caracterizan al AA demandan la aplicación de pensamiento de alto orden, y en muchos casos, exteriorizar lo aprendido. Esta exteriorización suele involucrar a otros estudiantes o instructores, y crea espacios para el desarrollo de competencias, meta-competencias y habilidades blandas. Estos factores son afines a la realidad que engloba a la IS, de donde se explica el interés en profundizar en el AA en IS.

2.9. Team-Based Learning

Team-Based Learning (TBL) es un método pedagógico activo y colaborativo. Se definen unidades temáticas y, para cada una, se trabaja en tres secciones: preparación previa, aseguramiento de la preparación y ejercicios de aplicación. TBL se apoya de manera mucho más fuerte en la interacción de pequeños grupos que otras estrategias de aprendizaje usadas en educación superior [43].

TBL se basa en cuatro elementos o pilares fundamentales: equipos apropiadamente formados, responsabilidad, feedback frecuente e inmediato y actividades de aplicación que promuevan tanto el aprendizaje como el desarrollo del equipo [44].

Equipos apropiadamente formados

Antes de comenzar la implementación de TBL se deben definir equipos. Estos equipos se mantienen a lo largo de todo el curso. El tamaño de los equipos puede variar, pero idealmente ronda los 5 a 7 estudiantes [44]. Los equipos los define el cuerpo docente; deben ser equipos equilibrados en factores que puedan determinar el rendimiento de los equipos [44]. Factores candidatos pueden ser barreras de idioma, experiencia laboral, rendimiento académico, entre muchos otros.

Por otro lado, deben evitarse equipos cohesivos por sinergias previas existentes que puedan condicionar las interacciones dentro del grupo, como relaciones familiares, grupos de amigos, parejas, entre otros. Si bien la diversidad dentro del grupo puede generar tensiones a corto plazo, a largo plazo esta diversidad se convierte en capital que enriquece al equipo [43].

Responsabilidad

La responsabilidad es un elemento fundamental de TBL: los estudiantes deben ser responsables individualmente ante el instructor y ante el resto de su equipo, y también deben ser responsables en equipo como una unidad. Los estudiantes son responsables de la calidad y cantidad de su trabajo tanto individual como colectivo.

Uno de los factores que más responsabilidad individual demanda es el RAP (Readiness Assurance Process o Proceso de Aseguramiento de Preparación). Una de las actividades de TBL es la clase invertida: los estudiantes se preparan antes de clase estudiando las unidades temáticas con material provisto por el instructor. Es responsabilidad de cada estudiante ir a clase preparado para llevar adelante las actividades de TBL que se presentan más adelante. También deben llevarse adelante evaluaciones entre pares para medir la contribución de cada miembro al equipo.

Feedback frecuente e inmediato

En TBL todas las actividades incluyen instancias de retroalimentación instantánea. La retroalimentación es fundamental para la retención y el aprendizaje en general, y además tiene gran impacto en desarrollo de equipos [43].

Actividades de aplicación que promuevan tanto el aprendizaje como el desarrollo del equipo

Las actividades de aplicación deben requerir interacción dentro del equipo: tomar decisiones exponiendo ideas simples y claras. Deben dispararse conversaciones valiosas sobre el tema a tratar y no sobre formatos o aspectos formales.

Actividades de TBL

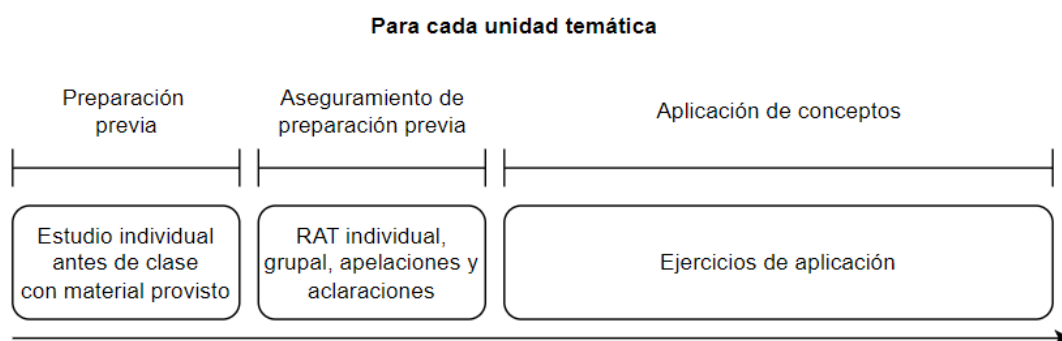


Ilustración 1: secuencia de actividades de TBL

La secuencia de actividades comienza con la definición de unidades temáticas principales, y para cada una se define el material de estudio. Previo a la clase, los

estudiantes se preparan consumiendo el material de estudio. El soporte de este material puede variar y ser combinado.

Al concurrir a la primera clase de una unidad temática comienza el RAP. Los estudiantes se enfrentan a una prueba individual. Esta prueba (individual Readiness Assurance Test o iRAT) es corta (5 a 20 preguntas de múltiple opción) y versa sobre las ideas principales del material de estudio. Inmediatamente, los estudiantes realizan la misma prueba en grupos (team Readiness Assurance Test o tRAT). Como el nombre lo dice, el propósito de estas evaluaciones cortas es darle información a los estudiantes y a los equipos acerca de qué tan bien preparados han concurrido a clases, y por tanto determinar en qué medida están listos para contribuir al trabajo en equipo. Las pruebas se realizan con tarjetas Immediate Feedback Assessment Technique (IF-AT) o plataformas digitales que proveen feedback inmediato.

El objetivo de repetir exactamente la misma prueba en equipo es brindar una nueva oportunidad de aprendizaje: puede ser que los estudiantes no hayan alcanzado por completo el aprendizaje esperado mediante la preparación previa. La instancia de RAT grupal permite intercambiar con los miembros del equipo para alcanzar un aprendizaje más completo apoyándose en equipo.

Tanto el iRAT como el tRAT impactan en la calificación de cada estudiante. Todo el RAP puede proveer entre el 15 y el 20% de la calificación de cada estudiante en un curso que se basa en TBL [45]. Algunos instructores permiten a los equipos determinar, dentro de ciertos parámetros, cuánto pesa cada tipo de RAT en cada equipo [43].

En la siguiente etapa del RAP, los estudiantes pueden escribir apelaciones basadas en evidencia si creen tener argumentos válidos para negociar respuestas que tuvieron mal. El instructor da una breve “lección” para clarificar cualquier problema de preparación que la clase pueda haber tenido.

El resto de la clase se usa en ejercicios de aplicación que promuevan el pensamiento de alto orden, y es para esta sección de la clase o las clases que se realizó con anterioridad el RAP.

3. Trabajos relacionados

Esta sección presenta documentos y trabajos anteriores que aporten a cumplir con los objetivos de investigación de conocer las estrategias de AA en IS (1) y conocer las experiencias de aplicación de las estrategias (2).

Varias universidades tienen unidades especializadas en educación que reportan estrategias de AA, como Iowa State University Center for Excellence in Learning and Teaching, Cornell University Center for Teaching Innovation y University of Minnesota Center for Educational Innovation.

Iowa State University Center for Excellence in Learning and Teaching ha publicado un documento con 226 estrategias de AA [46] generales, no necesariamente específicas para ingeniería o IS. Estas estrategias van desde hitos puntuales en el aula como escribir un Haiku (poema de tres líneas con 5, 7 y 5 sílabas respectivamente), hasta estrategias más amplias como generar comunidades de aprendizaje entre estudiantes. Cornell University Center for Teaching Innovation presenta 13 estrategias de AA para realizar en el aula y 3 para colaborar fuera del aula [47], también inespecíficas según disciplina. University of Minnesota Center for Educational Innovation presenta estrategias de AA según su complejidad y brinda ejemplos para grupos grandes [48].

En 2014 se realizó un mapeo sistemático sobre enfoques prácticos para enseñar IS [8]. Este estudio contempló 173 experiencias primarias de aplicación de técnicas de aprendizaje de IS, tanto activas como tradicionales, entre 1982 y 2013. La primera pregunta de investigación responde a cuáles son los principales enfoques para dar experiencia práctica a estudiantes en ingeniería de software. Aprender haciendo apareció en el 54% (93 experiencias), seguido por Project-Based Learning (PBL) en el 12% (21 experiencias), estudio de caso en el 9% (15 experiencias), juegos en el 7% (12 experiencias). Simulación, aprendizaje tradicional, open source, clase invertida y service learning aparecieron en 5% o menos del total de experiencias. La segunda pregunta de este mapeo sistemático apunta a responder si existe una tendencia para encarar el desafío de enseñar ingeniería de software de manera práctica. Los autores mencionan que los enfoques tradicionales (acompañar clases expositivas teóricas con algunas actividades prácticas) son los más usados porque son los más atractivos para los

instructores y han sido implementados por mucho tiempo. Es importante visibilizar los medios para enseñar IS de una manera activa para poder así satisfacer las necesidades que demanda la industria. Este estudio es relevante porque comparte algunos objetivos de investigación con el presente, y contempla un horizonte temporal complementario.

Una revisión sistemática de literatura acerca de educación en ingeniería de requerimientos [49] estudió un total de 152 publicaciones. Una de las preguntas de investigación refiere a cuáles son los enfoques educativos que se le da a la ingeniería de requerimientos y cuáles son los principales resultados esperados del aprendizaje. La revisión se centra en la agregación de resultados esperados en la formación en ingeniería de requerimientos y no profundiza en el enfoque pedagógico aplicado en cada experiencia. El estudio concluye afirmando que existe una tendencia hacia el aprendizaje basado en experiencias realistas en la industria para aprender ingeniería de software, y que el aprendizaje basado en teoría se encuentra en un segundo plano. También, presenta que en el área de ingeniería de requerimientos no se realiza una revisión sistemática o estudio tipo mapeo de literatura completo desde 2012. Finalmente, los autores mencionan que a futuro es necesario un estudio cualitativo para encontrar nuevos resultados. Este estudio es relevante porque explora estrategias de aprendizaje en un área específica dentro de la IS, y cubre un horizonte temporal que este estudio no.

En 2022 se realizó otro estudio secundario [50] sobre la efectividad de Project-Based Learning (PBL) para desarrollo de software. Este estudio reporta 54 publicaciones, de las cuales 20 fueron consideradas luego de aplicar criterios de inclusión y exclusión. Las publicaciones son estudios primarios de efectividad de PBL en desarrollo de software. La revisión sistemática concluye que la eficacia de PBL en ingeniería de software se observa en habilidades técnicas y habilidades blandas. La mayoría de las veces, la eficacia se evalúa mediante cuestionarios y plantean la necesidad de evaluar esta dimensión de otra manera. Este estudio es relevante porque sintetiza experiencias de aplicación de una estrategia específica relevante de AA como lo es PBL.

Otro estudio secundario [51] reúne experiencias de aprendizaje usando juegos en IS entre 1974 y 2016, donde se identifican 156 estudios primarios. Los autores concluyen que existen áreas de la ingeniería de software donde se puede ahondar más el uso de

juegos. Las áreas más exploradas en las experiencias son proceso de software, diseño de software y prácticas profesionales. Además, destacan la falta de sistematización del conocimiento acerca de los métodos de aprendizaje que utilizan juegos. Además, este estudio presenta distintas estrategias relacionadas a juegos como juegos serios, desarrollo de juegos y gamificación. Los resultados son relevantes pues sintetizan experiencias de aplicación de una estrategia específica relevante de AA como lo son los juegos.

El estudio presente complementa a los estudios previos presentados ya que indaga en las estrategias de AA en IS en general, sin limitar el área dentro de IS ni acotar a una única estrategia o familia de estrategias, y se enfoca en la actualidad, en particular durante y después de la pandemia causada a raíz de COVID-19.

4. Estrategias y experiencias de aprendizaje activo en ingeniería de software

4.1. Introducción

El propósito de esta sección es presentar las estrategias de AA en IS y sus experiencias de aplicación. La industria del software demanda profesionales con habilidades ya forjadas que difícilmente se obtienen en aulas que utilizan únicamente pedagogías tradicionales como las lecciones. Este conjunto de habilidades propias de una disciplina orientada a la industria y a la experiencia como la IS requiere situaciones de aprendizaje auténticas. Por esta razón es relevante conocer qué estrategias existen para crear situaciones de aprendizaje activo.

La educación en general, y en especial en IS, se vio disrumpida por la pandemia a raíz de COVID19. La educación remota e híbrida prepararon el camino para la transformación de la educación aún luego del retorno de la presencialidad. Este cambio forzó -y sigue forzando- a las instituciones educativas a transformarse.

La reconfiguración de la educación causada por la pandemia fue determinante a la hora de diseñar un método para cumplir con los objetivos de investigación del estudio presente. También lo fue el interés por parte de la Cátedra en aprender sobre las estrategias de AA en IS.

El primer objetivo de investigación (1) busca conocer cuáles son las estrategias de AA en IS en educación superior. El segundo objetivo de investigación (2) busca conocer cuáles son las experiencias de aplicación de estas estrategias entre 2018 y 2023.

El método elegido para llevar adelante esta investigación es de RR. La técnica de RR busca encontrar evidencia para la solución de un problema optimizando el tiempo y el esfuerzo. Las RR son estudios secundarios que emplean estrategias para acelerar el proceso tradicional de revisión sistemática [10].

Este estudio busca servir como un insumo más para la toma de decisiones en la Cátedra de Ingeniería de Ingeniería de Software de Universidad ORT Uruguay, y precisamente

esta es una ventaja de la RR contra la revisión sistemática de literatura: permite informar decisiones emergentes en contextos donde es necesaria la toma de decisiones veloz [52]. También ofrece la ventaja de permitir un fácil involucramiento de diversos interesados que llevan a la práctica el tema investigar, en lugar de interesados con perfil exclusivamente académico [53].

Una de las preguntas de investigación también invita a integrar literatura gris en el método. La literatura gris consiste en aquellos documentos que no son habitualmente encontrados en canales comerciales de publicación [54] y no suelen ser indexados. Esta parte del método es particularmente relevante para encontrar estrategias de AA que se apliquen de manera real, aunque no estén presentes en documentos de literatura académica. Las prácticas pedagógicas cambian muy rápido y no siempre es posible reflejar esta evolución en estudios, por ejemplo, arbitrados. Dada la naturaleza pragmática de los objetivos de este estudio, se contemplaron prácticas actuales de AA en universidades de renombre y de documentos de apoyo a la docencia de la Universidad ORT Uruguay.

Esta sección busca entonces identificar cuáles son las estrategias de aprendizaje activo en ingeniería de software utilizadas en educación superior y conocer las experiencias de aplicación de ellas. Para esto fueron elaboradas dos preguntas de investigación.

4.2. Preguntas de investigación (PI)

1. ¿Cuáles son las estrategias de aprendizaje activo usadas en ingeniería de software en educación superior?
2. ¿Qué experiencias de aplicación de las estrategias de aprendizaje activo usadas en ingeniería de software en educación superior existen?

4.3. Método

4.3.1. Búsqueda

Para la PI1 se realizaron dos actividades: por un lado, se visitaron los sitios de unidades especializadas en enseñanza de distintas universidades. Se visitaron Cornell University Center for Teaching Innovation, University of Minnesota Center for Educational

Innovation, The Derek Bok Center for Teaching and Learning de Harvard University, el sitio de AA del Massachusetts Institute of Technology y el Centro de Actualización para la Enseñanza Superior (CAES) de Universidad ORT Uruguay. Las estrategias reportadas mediante esta actividad no necesariamente son diseñadas en el contexto de cursos de ingeniería o IS, sino que son generales. La segunda actividad fue considerar todas las estrategias mencionadas en las experiencias relevadas para la pregunta de investigación (2) que involucra experiencias en IS o cursos afines a IS.

Para la PI2 se realizaron búsquedas en Portal Timbó (portal de Agencia Nacional de Innovación e Investigación (ANII) que provee acceso a buscador EBSCO Host) e IEEEExplore. En ambos portales se aplicó la cadena de búsqueda <<Active AND teaching OR learning AND software engineering>> en toda la metadata de las publicaciones.

Estas cadenas de búsqueda devolvieron un gran volumen de publicaciones sobre aprendizaje automático, donde aprendizaje activo tiene otro significado en el contexto de inteligencia artificial. En una segunda ronda, se refinó la cadena de búsqueda con la exclusión del término << machine>>. La cadena resultante fue <<Active AND teaching OR learning AND software engineering NOT machine>> en toda la metadata de las publicaciones.

El objetivo de la búsqueda es encontrar experiencias recientes, se consideraron estudios del 2019 a 2023. En Timbó el resultado inicial fue de 979000 publicaciones. Se tuvieron en consideración las primeras 60 publicaciones por orden de relevancia.

En IEEEExplore resultaron 13730 resultados, de los cuales se conservaron las primeras 75 publicaciones por orden de relevancia. Además, se aplicaron criterios de inclusión y exclusión.

Criterios de inclusión y exclusión

CI1. El estudio debe ser arbitrado

CI2. El estudio debe versar sobre educación en ingeniería de software en educación superior

CI3. El estudio, si es primario, debe ser una experiencia o propuesta de método

CI4. El estudio, si es secundario, debe ser sobre educación en IS

CE1. El estudio no ha sido publicado en 2018 o posterior, y en 2023 o anterior

CE2. El estudio no está publicado en español, inglés o portugués

Luego de aplicar los criterios de inclusión y exclusión, resultaron 33 estudios en IEEExplore y 16 estudios en Timbó. El número total de estudios relevantes encontrados es 49.

4.3.2. Extracción de datos

Para los sitios especializados en enseñanza de universidades se visitó el sitio web y se extrajo la información manualmente. Se extrajeron solamente las estrategias de aprendizaje que se mencionan en el sitio, y se ignoraron otros datos como recomendaciones o buenas prácticas.

Para los estudios sobre AA encontrados en los buscadores de investigación arbitrada se diseñó un formulario de extracción de datos. De cada estudio se extrajo: título, enlace, institución principal, tipo de estudio, volumen de experimento, año de publicación y técnicas de aprendizaje activo mencionadas.

Una vez identificadas las estrategias de aprendizaje activo se realizó una primera agrupación de técnicas con distinta denominación pero que comparten la misma base metodológica. Aquellas estrategias que compartieran muchos rasgos comunes (como Think-Pair-Share (TPS) y Think-Pair-Square o Game-Based Learning y el uso de juegos en general) fueron agrupadas bajo un único nombre (TPS y Think-Pair-Square fueron agrupadas bajo el nombre Think-Pair (TP), Game-Based Learning, Gamificación y juegos en general fueron agrupadas bajo el nombre juegos). Esta clasificación de estrategias bajo un mismo nombre aplica para las estrategias encontradas en estudios primarios.

4.4. Resultados

4.4.1. PI1: Estrategias de aprendizaje activo en ingeniería de software

Considerando todas las fuentes, se reportan en total 45 estrategias de AA, considerando solapamientos. 18 de ellas fueron identificadas mediante la RR. 36 de ellas fueron reportadas en páginas web de sitios especializados en enseñanza en universidades.

Las estrategias de AA reportadas en la RR son: juegos, Project-Based Learning (PBL), Flipped/Clase invertida, Peer-Based/Aprendizaje basado en pares (ABP), Problem-Based Learning (PrBL), debates, Multimedia, Think-Pair, Service Learning, Experiential Learning (ELT), Concept Test, Analogy, VR/AR, Team-Based Learning (TBL), Just In Time Learning (JIT), Crowd-Based Learning, Competence-Based Learning y Blended Learning.

En el sitio web <<Encouraging active learning in large class teaching>> de University of Oxford Centre for Teaching and Learning [55] se encontraron las estrategias: Think-Pair y mini-actividades.

En el sitio web <<Active learning>> de The Derek Bok Center for Teaching and Learning de Harvard University [56] se encontraron las estrategias: debate, discusiones, casos, trabajo en grupos, clase invertida, resolución de problemas, encuestas que puedan ser respondidas desde cualquier dispositivo inteligente (polling and clickers), toma de notas colaborativa, concept map, Jigsaw, Minute paper, Statement Correction/Intentional Mistakes y Strip sequence.

En el sitio web <<Active learning>> de Massachusetts Institute of Technology [57] se encontraron las estrategias: Concebir-Diseñar-Implementar-Operar, Aprendizaje Activo con Tecnología (Technology-Enabled Active Learning) y xTutor.

En el sitio web <<Active learning>> de University of Minnesota Center for Educational Innovation [48] se reportan las estrategias: Minute Paper, Pause for reflection, Self-assessment, Large Group Discussion, Triad Groups, Informal Groups, Peer Review, Group Evaluations, Brainstorming, Case Studies, Hands-on Technology, Interactive

Lecture, Games/Simulations, Role Playing, Inquiry Learning, Jigsaw, Forum Theater y Experiential Learning.

En el sitio web <<Métodos y técnicas de enseñanza>> del Centro de Actualización en la Enseñanza Superior (CAES) de Universidad ORT Uruguay [58] se reportan las estrategias: aprendizaje autónomo, el aprendizaje basado en problemas, nuevamente aparece el estudio de caso, proyecto y trabajo de campo.

4.4.2. PI2: Experiencias de aplicación de las estrategias de aprendizaje activo

Del total de estudios considerados, seis fueron discusiones de método, tres estudios secundarios y 40 experiencias, una de ellas inconclusa. Las experiencias son situaciones donde se aplicó una metodología o dinámica de AA en el contexto concreto de un curso o serie de cursos. El resto de los estudios elegidos son 6 propuestas de métodos para enseñar ingeniería de software de manera activa y 3 estudios secundarios.

En los estudios considerados por la RR se identificaron 18 estrategias de aprendizaje activo distintas. Entre ellas se encuentran métodos generales y otras técnicas puntuales. En la siguiente tabla se muestra la frecuencia de aparición de cada una.

Estrategia	Frecuencia
Games	16
Project-Based Learning (PBL)	14
Flipped	6
Peer-Based	6
Problem-Based Learning (PrBL)	5
Debate	4

Multimedia (MM)	3
Think-Pair (TP)	2
Service Learning	2
Experiential Learning (ELT)	2
Concept Test	2
Analogy	2
Virtual/Augmented Reality (VR/AR)	1
Team-Based Learning (TBL)	1
Just In Time Learning (JIT)	1
Crowd-Based Learning	1
Competence-Based Learning	1
Blended Learning	1

4.5. Discusión de estrategias de aprendizaje activo

Esta sección busca analizar los resultados obtenidos para las preguntas de investigación sobre cuáles son las estrategias de AA en IS y sus experiencias de aplicación. En primer lugar se desarrolla una definición de cada estrategia de AA. Luego se clasifican y analizan las estrategias según criterios prácticos que pueden ser utilizados por docentes para discutir y eventualmente aplicar las estrategias en su diseño instruccional.

Se busca determinar qué profundidad tiene el impacto de la aplicación de las estrategias en un diseño instruccional. Para ello se clasifica cada una de las estrategias relevada mediante la RR como un marco pedagógico o una técnica de AA. Luego se discute la naturaleza de las actividades que lleva adelante la estrategia, al determinar si estas actividades son propias de procesos de ingeniería de software o actividades de formación académica. También se discute la medida en la cual las estrategias son individuales o grupales. Finalmente se presenta la medida en la cual se combinan estrategias de AA.

4.5.1. Introducción a las estrategias de aprendizaje activo

A continuación se presenta una definición general de las estrategias de AA encontradas. Los términos o nombres de cada estrategia se mantienen en inglés por ser los más frecuentes en la literatura de educación en la ingeniería.

Project-Based Learning

Este tipo de aprendizaje comprende mejorar el entendimiento de los estudiantes sobre un tema completando proyectos significativos y desarrollo de productos [59]. Entre las características que mencionan sobre PBL se encuentran el estudiante participando activamente con el docente actuando de facilitador, es orientado a un producto particular y promueve la colaboración entre estudiantes. Una importante diferencia entre PBL y PrBL es que en PBL se aplica conocimiento previamente adquirido y el foco principal de la metodología está en generar un producto final [60].

En el contexto de IS, PBL tiene relevancia por reflejar el trabajo en equipos multidisciplinarios y permitir la posibilidad de construir un producto, resultado del proyecto. Este tipo de actividades enmarcan situaciones en las cuales habilidades técnicas y competencias pueden aflorar y ser volcadas al mismo tiempo.

Problem-Based Learning

El aprendizaje basado en problemas (PrBL) se caracteriza por pequeños grupos de trabajo de estudiantes y un tutor o facilitador, quienes trabajan en conjunto para dar solución a problemas. El facilitador impulsa a los estudiantes a identificar problemas,

conocer conceptos clave, investigar y compartir posibles soluciones, y reflexionar sobre el trabajo realizado [61]. La discusión en torno a PrBL suele involucrar la importancia del conocimiento del tema por parte del tutor, su capacidad de empatizar y facilitar el trabajo de los equipos, y el compromiso por parte de los estudiantes [62]. En PrBL el foco está en el proceso de alcanzar soluciones y alcanzar conocimiento que no fue formalmente transmitido con anterioridad por encima de generar un producto final [60].

Los problemas como técnica de aprendizaje favorecen el pensamiento divergente y crítico. Más aún cuando existe un factor de colaboración en el proceso. Al navegar el espacio de soluciones posibles para un problema los estudiantes tienen la posibilidad de recorrer un camino distinto a sus pares o a los demás grupos de trabajo. Esta manera de abordar el aprendizaje refleja la naturaleza constructiva del aprendizaje de IS.

Service Learning

Service learning es un enfoque pedagógico [63] que busca crear espacios para que los estudiantes reflexionen de manera crítica acerca del servicio a la comunidad en el contexto de un curso académico [64]. Los elementos que caracterizan a Service Learning son la participación en una actividad organizada que satisface necesidades previamente identificadas en la comunidad y la reflexión sobre la actividad tal que se logre comprender el contenido curricular, una óptica más amplia de la disciplina y un incremento en la responsabilidad cívica. Todo esto debe ocurrir obteniendo crédito académico [65].

Service Learning brinda un marco en el cual se pueden aplicar otras estrategias de aprendizaje a la vez que se vuelca el conocimiento y el esfuerzo a contextos auténticos. Si bien estos contextos pueden ser difíciles de encontrar en el mundo de IS, sumergirse en ellos puede brindar una de las experiencias más cercanas a la práctica profesional posibles.

Experiential Learning

Experiential learning es un tipo de aprendizaje en el cual quien aprende está directamente en contacto con la realidad estudiada [66]. Busca contrastarse con aquel aprendizaje en el cual el estudiante lee, escucha, escribe, pero nunca entra en contacto

real con el fenómeno a estudiar. El énfasis de este enfoque está en la experiencia sensible y las acciones contextuales [67]. La dialéctica de experiential learning se basa en la noción de que el conocimiento se crea mediante la transformación de experiencia.

La experiencia como camino al conocimiento no es una idea nueva. Experiential Learning provee un marco filosófico afín al aprendizaje activo, y en particular al inductivo. Es otro marco pedagógico mediante el cual se busca poner en contacto a quien aprende con contextos auténticos que puedan servir como antecedentes para enfrentarse a situaciones nuevas.

Just In Time Learning

Just In Time Learning (JIT Learning) no refiere a ninguna teoría o método de aprendizaje en particular [68]. JIT Learning, en contraste con el aprendizaje tradicional, tiene que ver con aprendizaje informal, guiado por el estudiante y enfocado en la aplicación del conocimiento. Este tipo de aprendizaje no está controlado o pre-diseñado. Los estudiantes definen el entorno de aprendizaje y lo adaptan según sus necesidades.

El aprendizaje JIT también involucra la noción de contextos auténticos. La idea de JIT vuelve a la naturaleza aplicada de la IS y el aprendizaje constructivo. Además, tiene la característica de impulsar al estudiante a conocer y a decidir sobre su entorno de aprendizaje, lo que lo vuelve un participante activo y responsable por su propio proceso.

Crowd-Based Learning

Crowd-Based Learning tiene como objetivo aprovechar el conocimiento de masas o multitudes (crowds) en internet para potenciar el aprendizaje [69]. Este marco pedagógico busca también lidiar con bloqueadores del aprendizaje existentes en contextos educativos tradicionales, como lo aislado de las organizaciones educativas, la limitación de fuentes de información. Fuentes de información notables en este contexto son YouTube, GitHub, StackOverflow, DockerHub y HuggingFace, entre muchas otras en auge por la comunidad Open Source.

Crowd-Based introduce un elemento característico de la práctica profesional de IS: el conocimiento disponible públicamente. Introducir este elemento en la formación de

futuros profesionales del software les permite introducirse al mundo profesional con una competencia -o metacompetencia- fundamental como lo es la capacidad de ubicar, valorar y comprender el conocimiento open source.

Competence-Based Learning

El aprendizaje basado en competencias es una metodología basada en el conocimiento que pretende demostrar de una manera más clara qué puede un individuo realizar como resultado del aprendizaje. En lugar de enfocarse en una porción de conocimiento como resultado de cada tarea, cada tarea es analizada para aplicar varias habilidades requeridas por la actividad [70]. CBL alienta a los educadores y estudiantes a ampliar la experiencia de compartir el conocimiento integrando los artefactos generados por masas en internet. Esto puede enriquecer el proceso de desarrollo de software de los estudiantes.

El aprendizaje basado en competencias tiene una arista común con el AA: el trabajo en el pensamiento de alto orden. Además, en este tipo de aprendizaje se valoran las competencias por encima del conocimiento abstracto. Este elemento recuerda al contexto en el que se da la ISA y los requerimientos que le propone la industria del software a los futuros profesionales.

Blended Learning

Blended Learning se caracteriza por la integración coordinada de métodos instruccionales tanto presenciales (cara a cara) y online, así como guiados por el instructor (síncronos) y al propio ritmo del estudiante (asíncronos) [71]. Este marco combina las fortalezas de las dinámicas presenciales con la conveniencia de las dinámicas remotas, y de este modo ofrece flexibilidad y diversidad tanto a la experiencia estudiante como a la docente [72].

Blended propone un aprendizaje adaptado a la realidad que vivimos. Este tipo de aprendizaje toma elementos propios de la realidad postpandemia y permite a docentes y estudiantes tomar decisiones sobre el proceso de aprendizaje: característica que, desde la óptica del estudiante, lo puede convertir en activo.

Team-Based Learning

Team-Based Learning (TBL) es una estrategia de aprendizaje activo que se enfoca en la aplicación de conocimientos mediante una secuencia de eventos que involucran trabajo individual, en grupo y con retroalimentación instantánea. Tiene tres etapas: la primera es la preparación anterior a la clase realizada por los estudiantes. La segunda es el aseguramiento de la preparación (readiness assurance) compuesta por pruebas individuales, grupales, apelaciones respecto a las pruebas y retroalimentación del instructor. La última etapa consiste en aplicación de conceptos clave [11].

TBL pone el centro en los equipos de trabajo y en la responsabilidad: características determinantes para los procesos de IS. Además, recoge otras técnicas de AA como flipped y peer-based.

Peer-Based Learning

Peer-Based Learning es una práctica educacional en la cual los estudiantes interactúan entre ellos para alcanzar determinados objetivos de aprendizaje [73]. Existen matices con el aprendizaje colaborativo, el cual reúne distintos marcos educativos que implican esfuerzo conjunto. Un caso particular de aprendizaje colaborativo es el aprendizaje cooperativo, en el cual, a pesar de que los estudiantes trabajen en grupos para alcanzar un objetivo común, el docente no deja de tener el control completo de la clase [74].

Al aprendizaje basado en pares reúne el elemento colaborativo de la IS así como el desarrollo de competencias. Muchas técnicas y marcos pedagógicos toman elementos de Peer-Based para mejorar el aprendizaje que de otro modo sería individual.

Juegos

Existen varios conceptos que involucran juegos en educación. Uno de ellos es el Aprendizaje Basado en Juegos o Game-Based Learning (GBL). GBL involucra diseñar e incorporar contenidos educativos en formato de juego, donde los jugadores participan activamente e interactúan con las mecánicas del juego para aprender y desarrollar habilidades [75].

Los juegos tienen cada vez más relevancia en educación. El juego es una actividad en sí misma, que además motiva y captura a los estudiantes. Es una estrategia de AA muy versátil, ya que es sencilla de integrar en cursos como técnica, así como también es posible direccionar un curso mediante juegos.

Flipped

Clase invertida es una experiencia de aprendizaje caracterizada por realizar las actividades que tradicionalmente se hacen en el tiempo y espacio de clase fuera de clase y viceversa [76][77]. Alienta a que los estudiantes avancen a su propio ritmo y fomenta el aprendizaje crítico y la discusión entre pares. Otro aspecto fundamental que hace a Flipped es la inversión del rol docente y estudiante. El conocimiento ya no se limita a lo que tradicionalmente el docente decide presentar en clase.

Flipped es una estrategia centrada en el estudiante: el estudiante es responsable de su aprendizaje. Poner al estudiante en el centro es intrínsecamente activo. Si bien implica esfuerzo adicional fuera de clase, permite generar dinámicas únicas en el contexto del aula.

Debates

Los debates tienen como propósito permitir a grupos de personas a exponer soluciones a problemas conociendo sus ventajas y desventajas. Los debates van de la mano con las discusiones. Estas exposiciones se realizan con un propósito de aprendizaje y suelen involucrar profundamente a los estudiantes. Esto genera una dinámica centrada fuertemente en el estudiante [78].

El debate está presente en muchas esferas de la vida de las personas. Apropiadamente implementado permite practicar pensamiento de orden superior de manera colectiva. Nuevamente aparece el pensamiento divergente como elemento axial.

Contenido multimedia

El contenido multimedia consiste en la presentación de información en más de un soporte. El uso de múltiples soportes para presentar la información hace a las prácticas instruccionales más efectivas y le da la posibilidad a los estudiantes e instructores de

elegir. Aún más cuando estos soportes son interactivos; es decir que están sincronizados de manera espaciotemporal [79]. Un ejemplo de soporte o medio son los videos. Los videos junto con actividades interactivas, tales como comentarlos o responder preguntas sobre ellos, constituyen una dinámica llamada Active Video Watching [80]. El uso de distintos medios de soporte para la información no necesariamente construye una dinámica de aprendizaje activo; deben existir actividades que involucren al estudiante con el material consumido.

El uso de contenido multimedia y la libertad para los estudiantes es un enfoque muy constructivista y conectivista [81] del aprendizaje. De manera estructurada puede promover la diversidad de fuentes de aprendizaje, previamente analizadas y criticadas. Además, activamente incluir estos elementos en procesos de aprendizaje contempla la realidad que rodea a los estudiantes en educación superior actual.

Think-Pair

Think-Pair (TP) es una dinámica que consiste en los estudiantes realizando una pregunta o resolviendo un problema en etapas. Primero individualmente y acto seguido, comparten su progreso con un par. Finalmente, comparten su progreso con toda la clase [82]. Existen variantes de esta técnica como Think-Pair-Square donde la tercera palabra indica que los estudiantes deben reunirse en grupos de aproximadamente cuatro a compartir su progreso, en lugar de hacerlo con toda la clase.

TP recoge ideas del aprendizaje basado en pares y estructura la estrategia. También involucra el pensamiento divergente, la colaboración y las competencias. Es un enfoque estructurado para bajar a tierra ideas propuestas por varios marcos pedagógicos.

Concept Tests

Los Concept Tests son pruebas cortas, informales y específicas que se realizan durante una clase para que los instructores evalúen si los estudiantes comprenden los contenidos fundamentales de alguna unidad temática. Su objetivo es obtener una idea general de qué tanto ha entendido la clase en su conjunto; no estudiantes en particular [83]. Los Concept Tests brindan una oportunidad para activar aprendizajes y recibir feedback inmediato.

Los Concept Tests son técnicas poco costosas y sencillas de implementar. El feedback inmediato es un elemento que sencillamente invita al AA.

Aprendizaje por analogía

El aprendizaje mediante ejemplos o analogía permite esconder los detalles específicos de ciertos contenidos y de tal modo facilita extraer conceptos comunes. Las analogías permiten relacionar un dominio desconocido con uno conocido y permitir más sencillamente la transferencia del conocimiento [84]. Esta herramienta debe ser diseñada con particular cuidado para evitar que los estudiantes caigan en interpretaciones erradas.

El aprendizaje por analogía pretende ser activo en tanto y en cuanto involucre pensamiento de orden superior en torno a los ejemplos o analogías. Además, se apoya fuertemente en el conocimiento previo: característica fundamental del aprendizaje significativo.

Virtual Reality/Augmented Reality

La realidad virtual involucra representaciones simuladas de apariencia real de un fenómeno. El propósito es sumergir al usuario en él. Habitualmente se alcanza mediante tecnología. Estos recursos se pueden integrar en entornos educativos para lograr experiencias educativas similares a la experiencia [85].

El uso de VR/AR aún tiene camino que recorrer; pero tiene sentido tomar esta técnica como una manera de construir contextos casi auténticos. También se apoya en la experiencia sensible y contextual.

4.5.2. Marcos pedagógicos y técnicas de aprendizaje activo

Es de interés clasificar las estrategias de AA según varios criterios. Uno de ellos tiene que ver con el involucramiento de la estrategia en un curso o el diseño del curso. Para ello se introducen dos ideas: los marcos pedagógicos y las técnicas. Los marcos pedagógicos son conjuntos amplios de dinámicas y abordajes filosóficos aplicados a lo

largo de un curso. Las técnicas son dinámicas puntuales de aprendizaje y son hitos individuales (aunque repetibles). Algunas estrategias pueden clasificarse en ambos grupos de acuerdo con el diseño instruccional.

Los marcos pedagógicos ofrecen la posibilidad de diseñar un curso con una filosofía de AA que lo respalde, pero es considerablemente más complejo de implementar. Las técnicas, en cambio, pueden entretenerse en cursos previamente existentes de modo que se agreguen estrategias activas de manera selectiva y gradual. Un ejemplo de aplicación de una técnica de AA en un curso ya existente podría involucrar un tema donde existan muchos puntos de vista y se lleve adelante un debate. Y quizás en unidades temáticas menos controversiales se lleven adelante actividades tradicionales como lecciones o clases magistrales, tal y como previamente se desarrollaba el curso. En la medida que la estrategia activa arroje buenos resultados, puede paulatinamente incorporarse otra técnica de AA, o no.

A continuación, se discuten las estrategias relevadas en la RR. Las estrategias relevadas en los sitios especializados en enseñanza clasifican todas como técnicas.

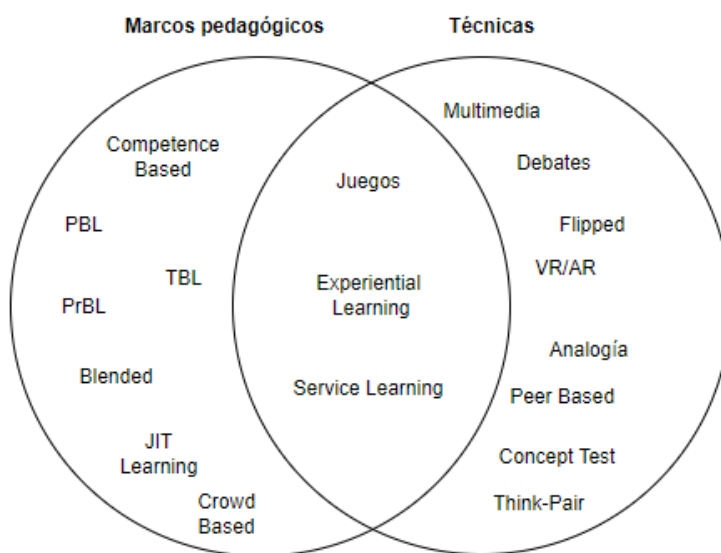


Ilustración 2: clasificación de estrategias según marco pedagógico o técnica

Existen marcos pedagógicos íntimamente ligados a estrategias. Por ejemplo, las estrategias proyecto y resolución de problemas serían clasificadas como técnicas. En cambio, las estrategias PBL y PrBL son clasificadas como marcos pedagógicos. El proyecto es perfectamente válido como hito puntual en un curso, pero implementado de tal forma no constituye una experiencia de aplicación de PBL.

Los cursos pueden incluir proyectos o problemas ocasionalmente, y por tanto diversificar y enriquecer la propuesta pedagógica del curso. En cambio, cuando se busca aplicar el marco pedagógico asociado a esas técnicas, el diseño instruccional cambia y se ve condicionado por el marco pedagógico. En ese caso, estas técnicas se convierten en directrices para el diseño instruccional: el curso se diseña a partir de las estrategias, mientras que cuando se usan ocasionalmente, las estrategias se diseñan a partir del curso.

Del mismo modo se puede discutir la naturaleza del uso de juegos contra Game-Based Learning (GBL). En GBL, el curso es diseñado a base de juegos tal y como el nombre de la estrategia lo indica. Sin embargo, existen cursos que no son diseñados a partir de juegos pero que aun así los utilizan como estrategia de enseñanza. Otra estrategia relacionada es la gamificación, que consiste en tomar una actividad y volverla un juego. Esta familia de estrategias fue agrupada toda bajo el nombre juegos.

En las experiencias relevadas, juegos aparece usada de ambas maneras, y por eso cae en la intersección. Service Learning y Experiential Learning se encuentran en una situación similar: existen cursos moldeados a través de estas estrategias, como existen intervenciones puntuales.

Estrategias generales y de aplicación de actividades de desarrollo de software

Las estrategias relevadas pueden involucrar actividades típicas del ciclo de vida de software como elicitación de requerimientos, desarrollo, prueba, validación y documentación. Otras estrategias pueden involucrar otro tipo de actividades especialmente valiosas en el contexto educativo pero que no son estrictamente

actividades vinculadas al desarrollo de software. Ejemplos pueden ser la reflexión, la auto-evaluación, la síntesis y transmisión de ideas, entre muchas otras. Esta clasificación puede ayudar a docentes a decidir qué estrategia usar según resultados de aprendizaje.

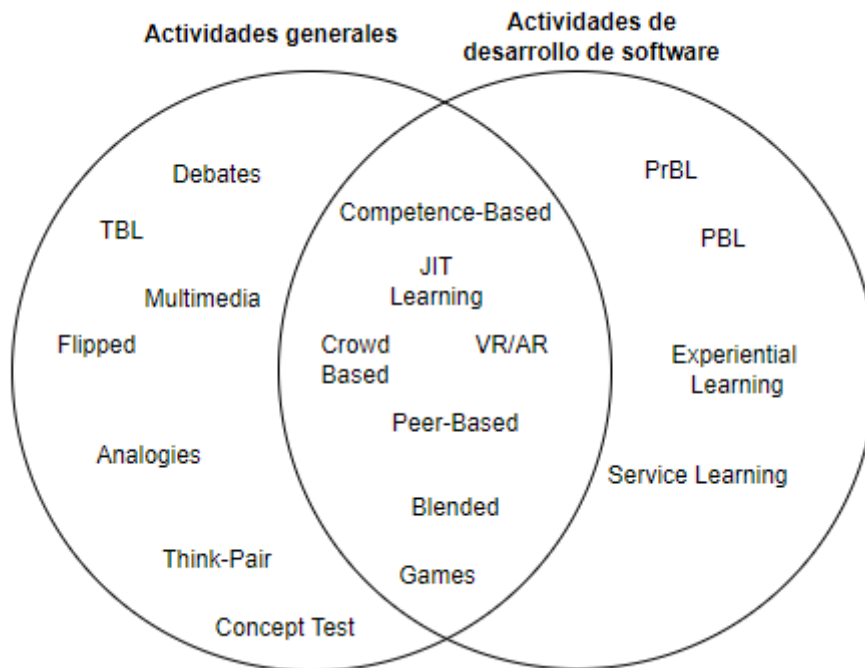


Ilustración 3: clasificación de estrategias según tipo de actividad

Algunos de los marcos y técnicas reportados hacen énfasis en la realización de actividades de aplicación de habilidades de ingeniería de software como el desarrollo de software, mientras que otras se basan en actividades analíticas o académicas, y otras incluso conviven en ambas clasificaciones. Por ejemplo Service Learning, Experiential Learning y Project-Based Learning aplicados en IS implican alguna actividad de desarrollo de software, tal como desarrollo o pruebas. Otras como VR/AR, Multimedia y debates son ejemplos de actividades de tipo analíticas, pues involucran actividades como la visualización de contenido y discusión de este. Existen ejemplos como Problem-Based Learning y Think-Pair que pueden caer en cualquiera de las categorías, y depende del diseño instruccional sobre el cual se aplique la técnica.

Las estrategias académicas destacan el valor del conocimiento en el proceso de aprendizaje de los estudiantes, ayudan a mejorar habilidades relacionadas al ámbito

académico como la lectura, la redacción escrita y la reflexión, y ayudan a asentar bases sobre las cuales continuar construyendo el conocimiento. Las estrategias de aplicación permiten adquirir y mejorar competencias fundamentales para el desarrollo profesional y brindan experiencias en entornos cercanos a los que el estudiante se enfrentará en su trayecto profesional.

4.5.3. Estrategias grupales e individuales

Varios marcos y técnicas reportados proponen como necesario o deseable el trabajo en equipo. Team-Based Learning es un ejemplo donde el trabajo en equipo no es opcional. El grupo de estrategias en las cuales el trabajo grupal no es opcional son Project-Based Learning, Problem-Based Learning, Debates, Crowd-Based Learning, Peer-Based Learning y Think-Pair. Crowd-Based Learning es un ejemplo destacable donde si bien no se fijan equipos a nivel de curso, el estudiante forma parte de una red activa de conocimiento e intercambio. Las demás estrategias pueden o no ser grupales. En la RR no se identifica ninguna estrategia que sea estrictamente individual.

Las estrategias grupales permiten practicar habilidades y competencias intrínsecas a los equipos de trabajo, como la negociación, la comunicación y la empatía. Las estrategias que pueden ser implementadas de manera individual, en cambio, permiten a los estudiantes centrarse en su propio proceso y avanzar a su propio ritmo, con el foco en el aprendizaje personal y la reflexión.

4.5.4. Combinación de estrategias de aprendizaje activo

Los estudios reportados en la RR revelan que múltiples técnicas y marcos pueden ser combinados. Los marcos pedagógicos aparecen acompañados en la mayoría de las experiencias. Esto ocurre para PBL (14 ocurrencias en total, 8 combinados), Peer-Based (6 ocurrencias en total, 4 combinados), PrBL (5 ocurrencias en total, 3 combinados). Experiencial Learning, TBL, JIT y Blended aparecen combinados en la única experiencia que reporta cada uno. Los restantes marcos pedagógicos (Service Learning y Competence-Based) se reportan sin ahondar en otras dinámicas.

Hay combinaciones de técnicas que funcionan juntas como Peer-Based con Think-Pair por la afinidad entre ellas. Otras técnicas y marcos combinables se complementan al

cubrir carencias, como se reporta en la experiencia aplicando PrBL con Experiential Learning. Existen combinaciones menos evidentes, como juegos con PBL, que no están necesariamente relacionadas entre ellas, sino que conviven dentro del diseño instruccional de un curso.

4.6. Amenazas a la validez

La clasificación inicial de estrategias de AA relevadas a partir de la RR según marco pedagógico o técnica, grupal o individual, y académica o de aplicación fueron realizadas por un único autor. La clasificación inicial fue revisada por tutores y se discutió la creación de categorías emergentes (marco pedagógico o técnica, grupal o individual, general o desarrollo de software).

La agrupación de estrategias similares (como por ejemplo GBL y juegos como una misma estrategia) fue realizada por un único autor. Esta agrupación se realizó en el contexto de una RR que busca ser consumida en procesos de toma de decisiones de la Cátedra de Ingeniería de Software. Cada estrategia puede presentar un sinnúmero de matices, y considerarlo exhaustivamente constituiría una digresión. Del mismo modo, estos matices pueden determinar causas para clasificar una estrategia como marco pedagógico o técnica, o ambas según su implementación.

A modo de ejemplo, el uso de juegos implica aplicar juegos ocasionalmente en el diseño instruccional de un curso o unidad de un curso. La gamificación consiste en introducir características típicas de juegos en actividades que habitualmente no son juegos. Game-Based Learning implica diseñar un curso o unidad de un curso mediante la aplicación de juegos. Las tres estrategias son distintas. Sin embargo, en este estudio todas serían mapeadas a juegos.

5. Aprendizaje activo en ingeniería de software en Universidad ORT Uruguay

5.1.Introducción

La Cátedra de Ingeniería de Software de Universidad ORT Uruguay busca enseñar habilidades de IS de manera significativa en contextos auténticos. Para llevar esto adelante, se aplican prácticas dentro y fuera del aula como clases invertidas, ejercicios de desarrollo de software, los mini-proyectos, los trabajos en equipo y marcos pedagógicos como TBL. Todas estas prácticas son ajustadas periódicamente para cumplir mejor con los objetivos educativos de la academia y de la industria.

Estas necesidades por parte de la industria de software incluyen los rápidos cambios de tecnologías y la adopción de nuevos paradigmas y técnicas como la ISA o DevOps. Por otro lado, las necesidades propias de la academia y de Universidad ORT Uruguay abarcan el incremento de cantidad de docentes y estudiantes, la oferta de clases presenciales, remotas e híbridas al mismo tiempo, y los desafíos docentes como el tiempo invertido fuera del aula preparando actividades, realizando correcciones o dando feedback.

A los efectos de continuar avanzando y adaptándose es importante reflexionar y mirar el camino recorrido. Gran parte del conocimiento acerca de la evolución del proceso de la Cátedra y sus actividades es tácito; es sabido por quienes han formado parte de esa historia pero no está sistematizado en el conocimiento organizacional.

Además, es valioso contribuir al proceso de sistematizar y disponibilizar este conocimiento. Las lecciones aprendidas pueden servir como insumo determinante en la toma de decisiones acerca de cursos brindados por la Cátedra.

El objetivo de este capítulo es presentar y reflexionar acerca de las diferentes actividades y evaluaciones llevadas adelante en materias de la Cátedra para formar profesionales con una visión integral en la IS de hoy en día. A lo largo del camino se

llevaron adelante varias actividades para enseñar IS, en particular utilizando estrategias de AA.

La principal aplicación de AA en la Cátedra son los obligatorios. Los obligatorios están presentes en todas las materias. Otra de ellas, TBL, refleja varios elementos de la forma en la que la Cátedra entiende se debe enseñar la IS. Tanto los obligatorios, TBL como las demás aplicaciones de AA reflejan la visión de la Cátedra sobre la responsabilidad individual y colectiva, el trabajo en equipos multidisciplinarios y el desarrollo de competencias. Por esta razón, la sección presente expone las diferentes aplicaciones de AA en la Cátedra además de la evolución del despliegue de TBL en las distintas asignaturas de la Cátedra que lo han aplicado.

5.1.Contexto y definiciones previas

Se presentará el contexto en el cual se desarrollan las asignaturas y sus evaluaciones en la Facultad de Ingeniería de la Universidad ORT Uruguay, con énfasis en aquellas impartidas por la Cátedra de Ingeniería de Software.

Las asignaturas que se presentan en este documento son aquellas pertenecientes a la Cátedra de Ingeniería de Software, curriculares y que forman parte de los planes de estudios de las carreras universitarias Licenciatura e Ingeniería en Sistemas a julio de 2023. Esto incluye: Fundamentos de Ingeniería de Software (FIS), Diseño de Aplicaciones 1 (DA1), Diseño de Aplicaciones 2 (DA2), Ingeniería de Software Ágil 1 (ISA1), Ingeniería de Software Ágil 2 (ISA2), Arquitectura de Software (AR), Arquitectura de Software en la Práctica (ASP), Desarrollo de Productos de Base Tecnológica (DPBT), dos materias de Interacción Humano-Computador (IHC) y tres materias de habilidades blandas.

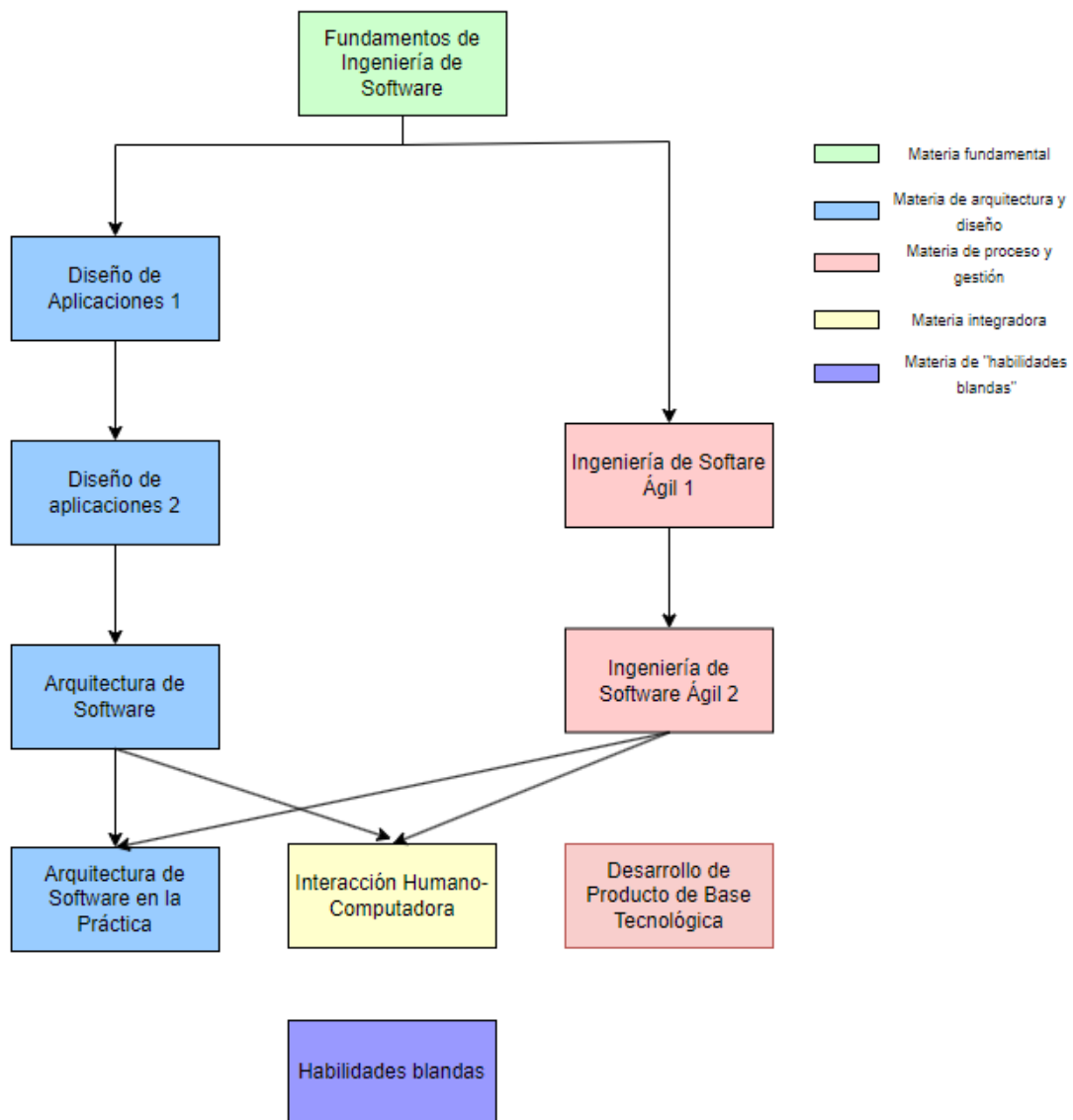


Ilustración 4: materias de la Cátedra y sus previaturas

Estas asignaturas pertenecen a las carreras Ingeniería en Sistemas (ID), Licenciatura en Sistemas (AN) o ambas. Las asignaturas de la Cátedra van desde el semestre 3 en AN y semestre 4 de ID con FIS, y cubren todos los semestres hasta el 7 en AN y el 8 en ID. Las materias pueden agruparse como materias de proceso o de gestión, las materias de diseño y arquitectura, materias integradoras y materias de habilidades necesarias para desarrollo de software (blandas). FIS es un caso particular porque actúa como fundamento para todo el resto de las materias. Es por esto por lo que en el orden de las asignaturas se observa una dependencia de FIS hacia todas las demás. Las materias que aparecen agrupadas horizontalmente se cursan en el mismo semestre.

En las materias de gestión entran ISA1 e ISA2. Estas materias introducen a los estudiantes a los aspectos de gestión en Ingeniería de Software con énfasis en el paradigma Ágil y equipos de trabajo multidisciplinarios.

Por otro lado, DA1, DA2, AR y ASP constituyen al grupo de las materias de diseño y arquitectura. Estas materias introducen y profundizan en el diseño de software con foco en el paradigma orientado a objetos (DA1, DA2), mientras que AR y ASP se enfocan en la arquitectura de software y ponen la lupa en las aplicaciones cloud-native.

Las materias de IHC son integradoras: buscan aplicar, mediante aplicaciones mobile, los conocimientos adquiridos tanto en materias de gestión como en materias de diseño y arquitectura. Las demás materias profundizan en habilidades no técnicas necesarias para la Ingeniería de Software, como las llamadas “habilidades blandas” y la gestión y diseño de producto.

Para enmarcar las estrategias de AA aplicadas en la Cátedra es relevante introducir las reglas generales para la aprobación y evaluación de las materias. A partir de los puntos asignados a cada actividad sobre 100 puntos se puede inferir directamente el impacto de esta en cualquier materia.

Las asignaturas de la Cátedra se aprueban con 70 puntos. Cualquier estudiante que alcanza ese porcentaje en la calificación final del curso se exime de cualquier evaluación posterior. Anteriormente, la mayoría de las asignaturas tenía un examen posterior a la finalización del curso en caso de que la calificación final del estudiante fuera mayor a 70 pero menor que 86. Este esquema de aprobación por examen post curso y posibilidad de exoneración del examen se dejó de utilizar en 2023.

En el caso que un estudiante termine un curso con una calificación menor a 70, si alcanza un puntaje mínimo en una evaluación determinada (de ahora en más, esta evaluación se dice que tiene una segunda instancia) y esa repetición puede suministrar un número tal que la nueva calificación final del curso sea 70, entonces deberá repetir tal evaluación, que puede ser cualquier evaluación (por ejemplo, en DA1 es el parcial).

En el caso que el estudiante termine el curso con una calificación menor a 70 y no alcance el mínimo en la evaluación que tiene segunda instancia, o no alcance el mínimo

en cualquier otra evaluación que tenga un puntaje mínimo, o la segunda instancia no pueda proveer un puntaje tal que la nueva suma de puntajes sea 70, entonces obtiene un cero en su promedio acumulado de calificaciones (PAC) y debe volver a cursar la materia.

Por ejemplo, si un estudiante obtiene 7 o más puntos en el obligatorio 2 de DA2, 9 o más puntos en el parcial, y el acumulado total de puntos es mayor o igual a 70, entonces aprueba. Si no obtiene 7 o más puntos en el obligatorio 2 o no consigue 9 puntos en el parcial, debe volver a cursar la materia. Si un estudiante obtiene 7 o más puntos en el obligatorio 2 de DA2, 9 o más puntos en el parcial, pero el acumulado total de puntos es menor a 70, entonces debe presentarse a la segunda instancia del parcial. Si en esa segunda instancia consigue un puntaje en parcial que al sustituir la anterior calificación de parcial permite llegar o pasar los 70 puntos, entonces el estudiante obtiene 70 como calificación final (sin importar cuánto fue el acumulado final) y aprueba la materia. Este flujo se explica gráficamente en el Anexo 2.

Diseño de aplicaciones 2 (6343)				
Ganancia de curso	SI		Obligatorio 1 - Sin defensa	
Examen posterior	NO		Puntaje máximo	20
Exoneración	NO		Puntaje mínimo	0
Obligatorio 2				
		Puntaje máximo	25	
		Puntaje mínimo	7	
Parcial				
		Puntaje máximo	45	
		Puntaje mínimo	9	
20% mínimo 2da instancia (9 puntos)				
Ejercicios (PSIO)				
		Puntaje máximo	10	
		Puntaje mínimo	0	

Ilustración 5: mecanismo de evaluación de DA2, recuperado de sitio de Bedelías

La evaluación de las asignaturas suele estar repartida entre obligatorios, parciales y ejercicios (PSIO: Parcial Sin Inscripción Obligatoria), según la tipificación de eventos de evaluación definida por la Facultad. Todas las asignaturas curriculares de la Cátedra de Ingeniería de Software dividen sus 100 puntos entre estas categorías.

A continuación, se presentan palabras propias de la jerga de Universidad ORT Uruguay. El propósito de esta lección es informar al lector externo a la Universidad y activar el conocimiento previo del lector familiarizado con los procesos y técnicas de evaluación que allí se llevan adelante. Estos conceptos serán clave a la hora de hablar de resultados y conclusiones.

Obligatorio

Los obligatorios son evaluaciones que pueden adoptar diversos formatos y ser realizados en equipos pequeños, típicamente en equipos autoseleccionados de 2 o 3 estudiantes. El propósito de este tipo de evaluación es incentivar a que los estudiantes se involucren activamente, resuelvan problemas y aprendan haciendo [86].

Habitualmente un obligatorio es un ejercicio práctico e integrador de mediano alcance, en grupos pequeños de dos o tres estudiantes (en casos excepcionales se realizan en equipos de más integrantes) con una fecha de inicio y fecha de finalización estipuladas donde se deben aplicar habilidades de desarrollo de software. La propuesta suele ser prescriptiva en el sentido que posee suficiente información para que los estudiantes comiencen a resolver el problema directamente. Existen oportunidades en las cuales los obligatorios parten de problemas abiertos también. A menudo, hay entregas de avance intermedias. Las tecnologías utilizadas en los obligatorios se actualizan para estar a la par con la industria. Ejemplos notables de tecnologías son JavaScript, Angular y Amazon Web Services.

Una tecnología habitualmente transversal a los obligatorios es GitHub. GitHub es usado desde FIS hasta materias avanzadas como ISA2 y ASP. Se utiliza para alojar código y artefactos de software en repositorios, para versionar, para introducir prácticas de calidad como Issues (informes de calidad, feedback técnico), Pull Requests (revisiones de código) y para realizar automatizaciones mediante GitHub Actions.

Parcial

Un parcial es una evaluación sincrónica y presencial, individual y en papel. Los parciales se diseñan contemplando estas restricciones. Por ejemplo, un ejercicio de desarrollo de software implica escribir el código en papel, y no se cuenta con un entorno

de desarrollo en computadora. Su duración típica es de 2 horas, aunque hay contadas excepciones. Las condiciones para los parciales están definidas por la Facultad de Ingeniería.

Ejercicios (PSIO)

Ejercicios (Parcial Sin Inscripción Obligatoria, PSIO) es un grupo de actividades flexible que puede ajustarse para la propuesta de cada materia. Cada asignatura y dictado (instancia puntual de una asignatura, con docentes y estudiantes específicos) puede llevarlos adelante de una manera distinta. Algunas asignaturas proponen tareas fuera del horario de clase, cuestionarios sincrónicos o asincrónicos, presenciales o no, en computadora, de manera oral o por escrito.

PSIO también puede considerar el puntaje de evaluaciones dentro del marco de TBL como RATs o ejercicios de aplicación. También puede integrar actividades como juegos, exposiciones orales e investigación de temas. La flexibilidad de los PSIO permite llevar adelante evaluaciones y actividades de marcos pedagógicos complejos y combinarlos: ejercicios de lectura previa, aplicación de conocimientos en equipo, entre otros, sin los requisitos formales que tienen los parciales u obligatorios (como el máximo de integrantes).

5.2. Pregunta de investigación

La pregunta de investigación principal es:

(3) ¿Qué actividades de aprendizaje se realizan en las asignaturas de la Cátedra de Ingeniería de Software de Universidad ORT Uruguay?

Uno de los objetivos de este estudio tiene que ver con la pregunta anterior y con poner bajo la lupa la evolución de la aplicación de TBL. La siguiente pregunta complementa a la ya presentada.

(3.1) ¿Cómo ha evolucionado la aplicación de TBL en las asignaturas de la Cátedra?

5.3. Método

El método para responder las preguntas de investigación involucra distintas actividades:



Ilustración 6: actividades para pregunta de investigación 3

En primer lugar, se consultaron documentos informativos y material de los cursos de la Universidad. Los documentos fueron comunicaciones oficiales de la Cátedra visibles para los estudiantes, como documentos de acuerdos de inicio de cursos y entradas en foros de consultas. También se recolectaron documentos provistos por Bedelías con los puntajes de cada evaluación en cada asignatura.

Para complementar el conocimiento de las evaluaciones y las asignaturas se realizaron entrevistas breves, puntuales e informales con docentes activos en aquellas asignaturas para las cuales no se conociera de antemano el funcionamiento y despliegue de las distintas evaluaciones.

Las evaluaciones fueron manualmente etiquetadas como activas o tradicionales, grupales e individuales por parte del autor de esta tesis, siguiendo un criterio definido y validado con tutores. El autor apeló al marco teórico y a su experiencia docente y estudiante para realizar esta clasificación.

Para responder a la pregunta de investigación específica (3.1) se realizaron entrevistas semi-estructuradas a seis docentes de la Cátedra con experiencia en asignaturas que apliquen TBL. Todas las entrevistas fueron realizadas entre octubre y noviembre de 2023 mediante videoconferencia o presencialmente. De las mismas se tomaron notas durante su ejecución.

Dos de los entrevistados son docentes de asignaturas con perfil técnico y se desarrollan hace años en la industria. Un entrevistado da clase hace más de una década y se ha desarrollado en la industria tanto de manera técnica como en aspectos de gestión. Tres

de los entrevistados son docentes activos hace décadas, son investigadores con amplia trayectoria en distintas áreas de ingeniería de software y se desempeñan en distintos aspectos de gestión y coordinación educativa.

El guion de las entrevistas semi-estructuradas se encuentran en el anexo (1). Las entrevistas fueron analizadas según las categorías de análisis: descripción de aplicación de TBL y su evolución, experiencia docente, percepción del docente de la experiencia del estudiante, desafíos, lecciones aprendidas y vistas a futuro.

En base a las notas tomadas durante las entrevistas, se agruparon por categoría de discusión pasajes, frases o combinaciones de palabras que remitan a la categoría. Por ejemplo, la frase <<Arrancó con la electiva metodologías ágiles/ISA (con juegos) aprox. 2015 y ya en 2016 se instaló en Arquitectura de Software>> se codifica en la categoría Descripción y evolución de TBL.

Otro ejemplo de codificación es la frase <<Empoderarlos y que tomen responsabilidad está bueno, pero hay otros donde al ser todo grupal no se sienten tan cómodos>> que entra en la categoría Percepción de la experiencia del estudiante.

En la categoría Lecciones Aprendidas aparecen frases como <<La preparación de la clase magistral es lo que más tiempo y esfuerzo consume>>.

5.4. Resultados

5.4.1. Evaluaciones y puntajes

En la ilustración 4 se puede visualizar la distribución de puntaje según activo o tradicional y grupal o individual para el acumulado de puntos obtenibles por estudiantes en todas las asignaturas de la Cátedra. El 55% del total de puntos obtenibles por estudiantes en las asignaturas de la Cátedra de Ingeniería de Software corresponden a evaluaciones grupales y de tipo activo. El 9% corresponde a evaluaciones individuales de tipo activo. El 36% restante corresponde a evaluaciones individuales de tipo activo. No se observan evaluaciones grupales de tipo tradicional (no activo).

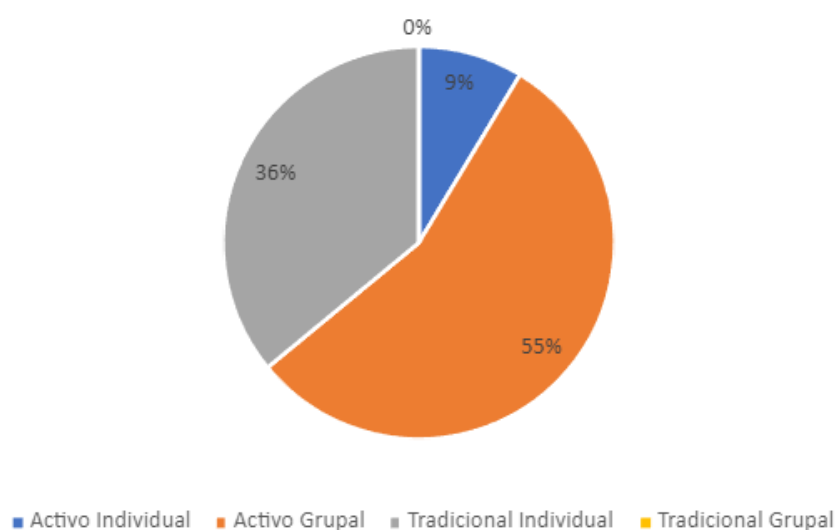


Ilustración 7: distribución de puntaje total en la Cátedra

En la ilustración 5 se puede observar la misma clasificación por materia, discernido según las dos posibles clasificaciones. En cualquiera de las dos clasificaciones la suma de los puntos en categorías complementarias suma 100, que es el total de puntos para la materia. Por ejemplo, en FIS el 40% de los puntos se obtienen en evaluaciones de tipo tradicional mientras que el 60% restante se obtiene en evaluaciones de tipo activo. Un segundo ejemplo de lectura de la tabla muestra que el 50% de los puntos de la materia DA2 viene de evaluaciones individuales, mientras que el 50% restante de los puntos viene de evaluaciones grupales.

	Individual	Grupal	Activo	Tradicional
FIS	40	60	60	40
DA1	50	50	50	50
DA2	50	50	50	50
ISA1	40	60	60	40
ISA2	48	52	80	20
AR	46	54	75	25

ASP	40	60	80	20
DPBT	40	60	60	40
IHC	30	70	70	30
DCU	50	50	50	50
TNEP	40	60	60	40
GCCM	40	60	60	40
HEDS	40	60	60	40
Cátedra	43%	57%	63%	37%

Ilustración 8: distribución de puntaje clasificado por materia y total relativo en la Cátedra

Puntos dentro de una misma evaluación pueden caer en diferentes categorías. Esto ocurre muy frecuentemente en PSIO. Por ejemplo, parte de los puntos PSIO de FIS viene de cuestionarios individuales en Aulas. Estos puntos de PSIO se clasifican como individuales y tradicionales. Otros puntos de PSIO en FIS vienen de ejercicio de aplicación de código. Esta parte de los puntos de PSIO de FIS clasifican como individuales y activos. Estos porcentajes reflejan la realidad relevada en julio de 2023, pero varían de manera frecuente de acuerdo con ajustes que se hacen a las evaluaciones.

5.4.2. Evolución de la aplicación de TBL

En las siguientes ilustraciones se pueden observar los principales hitos en la evolución de TBL en las asignaturas de la Cátedra que han aplicado TBL.

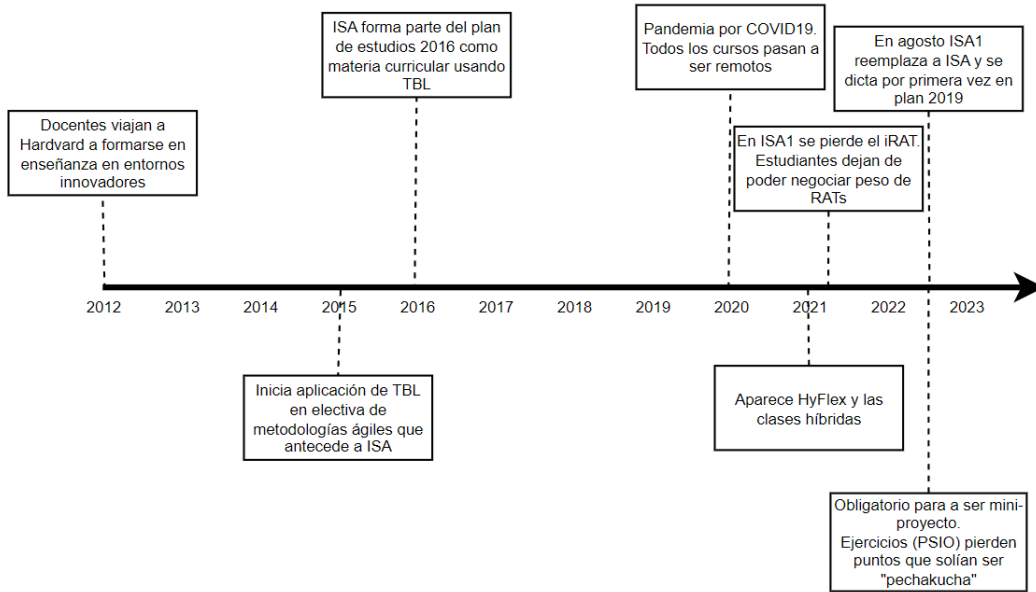


Ilustración 9: hitos relevantes en la evolución de ISA e ISA1

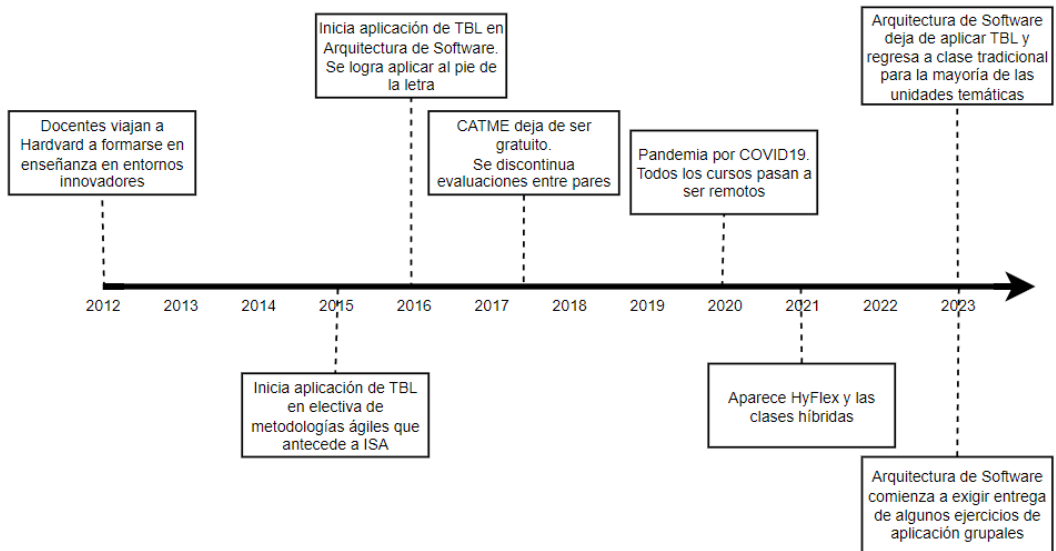


Ilustración 10: hitos relevantes en la evolución de AR

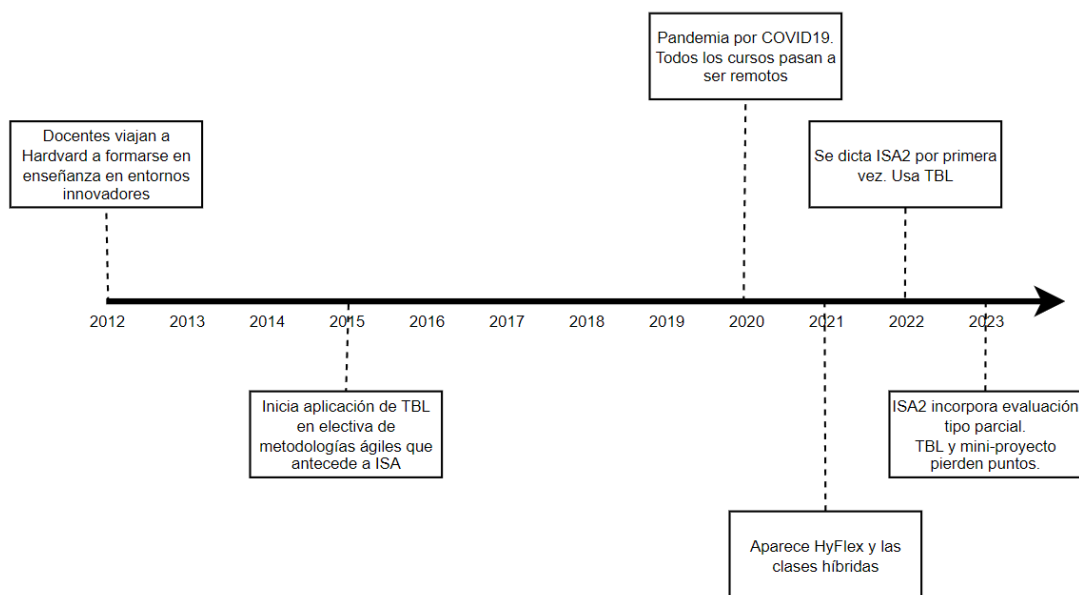


Ilustración 11: hitos relevantes en la evolución de ISA2

El involucramiento comenzó en 2012 con una instancia de formación en el exterior para algunos docentes en Uruguay sobre enseñanza en entornos innovadores: LASPAU. Algunos de los docentes de la Cátedra que participaron vieron con interés TBL. En 2015 comienza a aplicarse en una electiva de metodologías ágiles (materia que más adelante se volvería curricular: Ingeniería de Software Ágil (ISA) predecesora a ISA1). En 2016 se empieza a aplicar TBL en AR e ISA pasa a ser curricular: estas dos materias se convierten en las primeras curriculares en aplicar TBL. Para fin de ese año, la aplicación de TBL en AR es al pie de la letra.

En 2017 la herramienta CATME (entre otros usos se utilizaba para realizar evaluaciones entre pares de forma eficiente) deja de ser gratuita y se discontinúan las evaluaciones entre pares, por tanto, se deja de aplicar TBL al pie de la letra. Hasta 2020 no hay hitos determinantes: existen ajustes circunstanciales de puntajes y de operativa en algunas asignaturas. A nivel universidad, en 2019 se comienzan a dictar planes de estudio nuevos para ID y AN.

En 2020 Uruguay se ve afectado por la pandemia a raíz de COVID19. Esto obliga a volver todos los cursos remotos. Todas las actividades curriculares se ajustan para poder cumplir con la nueva modalidad. En 2021 la Universidad ORT Uruguay adopta el

sistema HyFlex para ofrecer clases híbridas. Además. En ISA se deja atrás el iRAT. De la mano con este cambio, los estudiantes pierden la oportunidad de negociar qué peso le dan a cada tipo de RAT, libertad que hasta entonces tenían (bajo ciertos parámetros).

En agosto de 2021 se dicta por primera vez ISA1, en lugar de la anterior ISA. El obligatorio de esta materia pasa a ser el mini-proyecto que se realiza actualmente. Esta evaluación pasa a tener un puntaje mayor y pierden puntos otras evaluaciones, como los ejercicios, que hasta entonces involucraban presentaciones con formato “Pechakucha”. En 2022 se dicta por primera vez ISA2, tercera materia curricular en incorporar TBL.

En 2023 AR deja de usar TBL en algunas unidades temáticas, pero mantiene algunos elementos de TBL en otras unidades. Esta materia también comienza a exigir la entrega de algunos ejercicios de aplicación para palear con el poco protagonismo que hasta entonces estaban teniendo. Finalmente, ISA2 incorpora una evaluación tipo parcial a costas de algunos puntos de actividades de TBL y del mini-proyecto.

5.4.3. Entrevistas con docentes

A continuación se presentan los resultados de las codificaciones de las respuestas a las entrevistas semi-estructuradas en base a las secciones temáticas ya definidas.

Experiencia docente

Los docentes relatan que aplicar TBL cambia radicalmente el rol docente. El docente deja de exponer información permanentemente y se vuelve un facilitador. El tiempo del aula se utiliza más para generar discusiones e interacciones significativas en lugar de relatar aquello escrito en libros. Los divierte y motiva más.

Comenzar a dar clase con TBL es costoso, pues hace falta preparar el material de lectura, cuestionarios y ejercicios. Luego, el esfuerzo se vuelve menor, en comparación con la clase tradicional. Preparar clases magistrales es lo que más esfuerzo lleva. A pesar del esfuerzo reducido, sigue existiendo esfuerzo fuera del aula: corrección y mantenimiento de las actividades y evaluaciones.

Percepción docente de la experiencia del estudiante

Los docentes expresan que el primer encuentro de los estudiantes con TBL genera dificultades para comprender la dinámica y para implementarla. A medida que avanza el curso se amoldan a ella. La mayoría de los cursos presenta quejas: algunas por el volumen del material de estudio y otras por la carga que TBL le propone al estudiante. Falta impulsar con más fuerza la visión de TBL.

Los docentes también observan que en TBL los estudiantes participan más y perciben que los estudiantes tienen más oportunidades de aprendizaje (preparación previa, RATs, apelaciones y ejercicios). Finalmente, los docentes transmiten que existe una brecha en la experiencia de los estudiantes que participan presencialmente de las clases respecto a los que participan de manera remota o directamente no participan.

Desafíos

La mayoría de los desafíos expresados por los docentes versan en torno a los ejercicios de aplicación: es difícil mantenerlos interesantes, actualizarlos, vincularlos con los proyectos en el caso que la materia tenga proyectos, y cómo mantener a los estudiantes participando de la clase, aunque los ejercicios de aplicación no siempre lleven puntos.

También transmiten desafíos en torno a la operativa de TBL. Es desafiante armar los grupos, los bancos de preguntas de los RATs, generar dinámicas valiosas en entornos híbridos, decidir el puntaje que lleva cada actividad y balancear el esfuerzo docente-estudiante.

Finalmente, expresan desafíos en torno a lo pedagógico. Por ejemplo, medir cuánto aprenden los estudiantes realmente y cómo es posible medir y mejorar el rendimiento de cursos de un semestre a otro.

Lecciones aprendidas

La primera lección aprendida es que las dinámicas activas, y en particular TBL, enriquecen los cursos de IS. La profundidad y amplitud de los temas ha incrementado positivamente con el tiempo. La segunda lección aprendida es que los estudiantes no valoran del mismo modo aquello que no impacta en su calificación.

Aplicando TBL puede darse la situación en la cual el docente se confíe que el estudiante aprendió, aunque no siempre sea así. Esto llevó a volver a valorar la lección o clase magistral para tener un nivel adicional de seguridad de que los estudiantes están siguiendo el curso.

Diseñar un curso mediante TBL puede presentar un desafío a la hora de conectar la tecnología estudiada en el curso con el material teórico. Finalmente, las dinámicas presenciales dan la sensación de brindar mejores resultados que las remotas o híbridas.

5.5. Discusión de actividades y evaluaciones en la Cátedra de Ingeniería de Software en Universidad ORT Uruguay

5.5.1. Actividades y evaluaciones en la Cátedra

La Cátedra heredó un legado de obligatorio como actividad formativa principal y con un peso importante en la evaluación. Es de esperar, entonces, que la mayoría de los puntos asignados a estudiantes vengan de evaluaciones grupales. Aproximadamente el 57% de los puntos totales vienen de evaluaciones grupales. El 43% restante viene de evaluaciones individuales.

Entre las evaluaciones individuales destacan los parciales. Los parciales son individuales, sin excepción. Otro volumen, menor, de puntos asignados individualmente vienen de ejercicios (PSIO) individuales. La mayoría de las veces, estos ejercicios individuales toman la forma de cuestionarios.

Las evaluaciones grupales son más heterogéneas. Los obligatorios son el principal insumo de puntos grupales. También hay puntos que vienen de los RATs grupales en los casos de TBL. Otros vienen de ejercicios (PSIO) grupales.

Existen desafíos que generan resistencia a virar hacia un mayor volumen de evaluaciones grupales. Uno de ellos es la contribución individual en las actividades grupales. También existen restricciones normativas de parte de la Universidad que limitan los tamaños de grupos de evaluación. En ISA1, por ejemplo, se realizan evaluaciones con gran número de estudiantes (hasta cinco o seis estudiantes por equipo)

en la modalidad de TBL y mini-proyecto, y el puntaje computa como ejercicios (PSIO) en ambos casos.

Otro elemento que llama la atención es la no existencia de puntos grupales-tradicionales. Este fenómeno tiene que ver con la clasificación realizada por el autor. Sin embargo, puede explicarse por la relación entre el aprendizaje cooperativo y el AA. De hecho, el aprendizaje cooperativo es un subconjunto del AA [87]. Tiene sentido entonces aseverar que, de acuerdo con el marco conceptual sobre el cual se desarrolla este estudio, las actividades cooperativas son activas. La cooperación entre miembros de equipo requiere interacciones significativas que activan el pensamiento de orden superior.

En contextos de aprendizaje cooperativo, el éxito del estudiante depende no solo de qué tan bien alcanza los resultados de aprendizaje individualmente; también depende de la medida en la que los demás miembros del grupo lo hagan. Esto refleja la filosofía de la Cátedra detrás de la IS: equipos multidisciplinarios trabajando en productos digitales.

Por otro lado, aparece el desafío de los “free riders” [87], es decir, estudiantes que se esfuerzan menos y se apoyan en el mérito de sus miembros de equipo. Esto, sumado a restricciones institucionales, han impulsado transformaciones que buscan ponderar con más fuerza el esfuerzo individual. Estas medidas se están implementando al momento de redacción de este documento.

El problema de la contribución individual en dinámicas grupales tiene valor desde una óptica acreditativa y desde una óptica pedagógica. El AA de IS en educación superior está enmarcado en instituciones educativas formales con procesos bien definidos. La integridad de las evaluaciones es fundamental para sostener la credibilidad y reputación de estas instituciones. La integridad de las evaluaciones también respalda la posibilidad de un aprendizaje auténtico e individual; pero con la capacidad de transferirlo a situaciones grupales o de equipo.

Otra arista por la cual es interesante analizar las evaluaciones y actividades en la Cátedra es la de activo y tradicional. Un elemento que surge tanto del análisis cuantitativo de evaluación como del análisis cualitativo a partir de los relatos docentes

es el interés en combinar estrategias de AA con estrategias de aprendizaje tradicional. Este elemento también está presente en trabajos relacionados ya presentados.

El AA es favorable para la enseñanza superior de IS, pero también existe un factor cultural en la educación superior que influye tanto en experiencia docente como en experiencia estudianta. Este factor es el que genera resistencia a la hora de introducir actividades que puedan disrumpir el proceder habitual de cursos de ingeniería. Este es otro argumento a favor de la combinación de estrategias activas con tradicionales; los cursos con dinámicas innovadoras, en especial en las que el rol docente cambia, generan opiniones por parte de los estudiantes. Esto aparece en las entrevistas con docentes al hablar del comienzo del despliegue de TBL, en especial en ISA1, que es la primera materia donde los estudiantes experimentan TBL.

Otra faceta de este elemento cultural es el valor que tiene el conocimiento per se en educación superior. Las estrategias activas tienen un foco en el hacer, muchas veces para hacer de nexo entre la academia y la industria. El aprendizaje tradicional, en cambio, tiene una manera de presentar el conocimiento de manera deductiva, partiendo de bases habitualmente teóricas que constituyen valor académico en sí mismas. Estas bases también permiten nivelar el aprendizaje entre estudiantes que construyeron conocimiento por caminos distintos. Estas ideas se han volcado a la práctica, por ejemplo, introduciendo una actividad de tipo parcial en ISA2.

5.5.2. Despliegue de Team-Based Learning

Casi una década después de incursionar en TBL por primera vez, la valoración global del uso de la estrategia es positiva. Dos de las tres asignaturas que aplicaron TBL mantienen su uso con ajustes. AR es un caso particular.

El caso de AR tiene que ver, por un lado, con la valoración de los docentes. Por otro, la arquitectura de software es una disciplina técnica y fundacional en la formación de ingenieros y licenciados en sistemas. Es una disciplina con bases teóricas muy bien asentadas que puede aflorar en clases de tipo expositivas. Naturalmente, esto debe darse en un contexto donde existan espacios para transferir el conocimiento a contextos

auténticos. Esto se ve ampliamente respaldado por las horas de “tecnologías” que tiene el curso de AR, que son horas de clase (la mitad de la carga semanal total) dedicadas exclusivamente a aplicar los conceptos teóricos mediante tecnologías y trabajar en el proyecto del curso (obligatorio). Este es un claro ejemplo de la coexistencia y complementariedad de estrategias de aprendizaje activas y tradicionales.

Una idea que aparece en las entrevistas con docentes de todas las asignaturas es el desafío que implica desplegar los ejercicios de aplicación en TBL. Habitualmente, el puntaje asignado mediante TBL a los estudiantes reside en los RATs. Los RATs son muy convenientes como mecanismo de evaluación por puntos. El problema aparece cuando los ejercicios de aplicación no llevan puntos. Muchos estudiantes, en especial en cursos nocturnos, se van de la clase cuando llega el momento de hacer ejercicios de aplicación que no son considerados en la calificación del curso.

Los ejercicios de aplicación son en realidad el núcleo de TBL: lo que viene antes es solamente una preparación para esta etapa de la clase. Algunos docentes expresan que impulsar la visión de TBL de manera fiel a como TBL es formalmente es una oportunidad de mejora. Y esta visión debe alcanzar tanto a docentes como a estudiantes; no alcanza con forzar la participación en ejercicios de aplicación mediante la adición de puntos por su realización, si bien es un buen primer paso.

También es importante hacer hincapié en cómo se vinculan las actividades de TBL con las actividades que no son de TBL. Un interesante ejemplo es ISA2. ISA2 tiene fuertes bases teóricas respaldadas por TBL y un parcial. Por otro lado, tiene un mini-proyecto que busca transferir esas bases teóricas a un contexto auténtico respaldado por tecnologías. Muchas veces estos componentes quedan débilmente ligados, o es difícil hacer explícita su relación. Esta es otra oportunidad de mejora expresada por docentes entrevistados.

Otro desafío es el de conjugar actividades tradicionales y de tipo activo. Es un desafío para todos los involucrados en el aula: desde la óptica docente las actividades de enseñanza tradicionales son más intuitivas de planificar, aunque requieren constante esfuerzo. Desde la óptica estudiante, son las actividades más fáciles de recibir. TBL no

es la excepción a ello. La introducción paulatina de nuevas dinámicas en el aula es un mecanismo importante para amortiguar el impacto de la disrupción en el aula.

5.5.3. Reflexiones de aprendizaje activo en ingeniería de software

La Cátedra hace uso de estrategias de AA así como de estrategias de aprendizaje tradicional. En el lenguaje de la Facultad de Ingeniería, los obligatorios casi siempre representan mini-proyectos y algunas veces problemas acotados. Los parciales reivindican la veta de aprendizaje tradicional que prevalece en educación terciaria. Los ejercicios son el espacio que más frecuentemente se usa para innovar en el diseño instruccional.

Los proyectos son la estrategia de AA más frecuentemente usada en la Cátedra. También es, junto con los parciales, la estrategia de aprendizaje que más puntos provee a los estudiantes.

Las decisiones en torno al diseño instruccional en la Cátedra no responden solamente a valoraciones pedagógicas. Existe un sinnúmero de factores que impactan en mayor o menor medida en las decisiones a nivel institucional, a nivel de Cátedra, a nivel de materia y a nivel de dictado. La pandemia causada por COVID19 fue uno de los factores con más fuerza.

En términos cuantitativos existe una tendencia en la Cátedra hacia actividades y evaluaciones de tipo activo. En las materias más avanzadas en la carrera como ASP e ISA2 esta tendencia se acentúa aún más. Esta tendencia invita a pensar en qué medida los estudiantes realmente aprenden haciendo, pues muchas veces las estrategias de AA dan mayor libertad a los estudiantes. También, la mayoría de las evaluaciones de tipo activo son grupales. Este es otro factor que pone tanto la integridad de las evaluaciones como el aprendizaje real de los estudiantes.

Por otro lado, esta tendencia también es un signo de que la Cátedra intenta proveer contextos auténticos para el desarrollo de habilidades de IS. Los profesionales de software trabajan en equipo y llevan adelante actividades de IS en el día a día.

Mirando hacia adelante, puede ser interesante sistematizar los procesos de transformación a nivel de Cátedra y de materia. Cada año aumenta el volumen de cursos, estudiantes y docentes. Este ritmo genera grandes volúmenes de información que pueden alimentar decisiones valiosas. Aún más sabiendo que muchas de las dinámicas que se llevan adelante en la Cátedra también se llevan adelante en otras instituciones en el mundo.

Sin embargo, existen algunas estrategias relevadas que no se han difundido con fuerza en la Cátedra, como TP, Service Learning y juegos. De acuerdo con las necesidades de cada materia, en particular el balance grupal-individual, académico o de aplicación y activo o tradicional, pueden valorarse individualmente. Eventualmente podrán adaptarse para enriquecer el repertorio pedagógico de los docentes.

El desafío no reside en la introducción de nuevas estrategias de AA en los cursos por el mero hecho de hacerlo. El desafío reside en balancear las estrategias de AA con las de aprendizaje tradicional para abarcar el pensamiento de orden inferior y superior. Y dentro de las de AA balancear las características discutidas anteriormente acerca de las mismas: la dificultad y costo de implementarlas, el ser grupales o individuales, el ser actividades de aplicación o de valor analítico y al fin y al cabo en qué medida son sostenibles en el tiempo para los docentes y estudiantes.

Puede ser valioso sistematizar la recolección de feedback por parte de los docentes para nutrir el proceso de transformación pedagógica. Los cursos son entidades vivas; años atrás las actividades de enseñanza tradicional eran la norma. Hoy existe un pivoteo hacia elementos como el AA, las competencias y los proyectos. Las distintas instituciones en el mundo perfilan sus actividades hacia lo que pueden hacer con los recursos que tienen, y mirar hacia el exterior siempre permite adquirir perspectivas nuevas.

5.6. Amenazas a la validez

La clasificación de puntos de evaluación en la Cátedra según individual o grupal, así como activo y tradicional puede variar de curso a curso con cierto margen.

Dada la realidad de los miembros entrevistados de la Cátedra, las notas de las entrevistas fueron tomadas durante la entrevista y de manera sintética. En la captura y

procesamiento de las entrevistas puede haber imprecisiones que impacten en los resultados presentados. La codificación temática fue realizada por el autor de esta tesis.

6. Conclusiones

Los trabajos relacionados arrojan que existen estrategias de aprendizaje activo, en auge en educación superior en ingeniería de software y que su aplicación es exitosa cuando son correctamente implementadas. Esta búsqueda identificó 18 estrategias de aprendizaje activo en ingeniería de software de aplicación reciente en educación universitaria (reportadas entre 2018 y 2023). El diseño instruccional puede enriquecerse de tomar estrategias de aprendizaje activo y combinarlas con estrategias de aprendizaje tradicional para generar experiencias en el aula más cercanas a la práctica profesional.

Algunas de las estrategias de AA relevadas son aplicadas en la Cátedra de Ingeniería de Software de la Facultad de Ingeniería de Universidad ORT Uruguay. Otras de las estrategias no han sido explotadas de manera general en las asignaturas curriculares de la Cátedra aún. Los proyectos son la estrategia de AA más aplicada en la Cátedra, habitualmente en la forma de obligatorios. Los obligatorios son proyectos de pequeña escala en un contexto tecnológico auténtico. También es una de las estrategias que más veces aparece en los estudios primarios relevados.

La Cátedra de Ingeniería de Software de la Facultad de Ingeniería de Universidad ORT Uruguay ha adaptado prácticas instruccionales, y en particular Team-Based Learning, en pos de mejorar la experiencia tanto docente como estudiante. Team-Based Learning es utilizado en la Cátedra de Ingeniería de Software hace casi una década y ha mejorado la experiencia tanto docente como estudiante en tres asignaturas.

Los docentes que han aplicado Team-Based Learning en sus cursos están satisfechos con su impacto en la experiencia docente y estudiante y han logrado adaptar el marco pedagógico a las necesidades de sus cursos. Aún existen desafíos intrínsecos al dinamismo de la vida académica como los métodos de evaluación y actividades a desarrollar dentro y fuera del aula que dirigirán muchas de las decisiones a corto y mediano plazo acerca del futuro de las asignaturas curriculares de la Cátedra. Ejemplos de estos desafíos son el costo de la preparación de clases y la receptividad percibida por los estudiantes.

6.1. Líneas de trabajo a futuro

Un trabajo complementario podría ser una revisión sistemática de literatura acerca de AA en IS. El proceso de RR sacrifica alcance y precisión por el afán de ganar tiempo y volcar resultados rápidamente en un contexto que los necesita.

El trabajo presente es exploratorio y descriptivo; podría ser de interés aplicar sus resultados en un caso de estudio o experimento controlado de aplicación de alguna estrategia de AA en cursos de la Cátedra y reportar la experiencia.

Una óptica poco cubierta por este estudio es la del estudiante. El único análisis que contempla la experiencia del estudiante en las asignaturas de la Cátedra es el realizado a partir de la percepción del docente de la experiencia del estudiante. Este trabajo debería ser realizado con mucho cuidado ya que los estudiantes persiguen los propósitos tanto de aprender como de aprobar, en distintas medidas.

También es de interés conocer qué estrategias dan los mejores resultados; es decir, valorar las estrategias reportadas.

Además, la información recaudada en este estudio sirve como insumo para un proyecto de mejora de TBL (y por tanto de la evaluación en general) de ISA2 en el semestre impar de 2024. También se propondrá un proyecto de fin de carrera de desarrollo de una plataforma de aprendizaje digital basado en los principios de aprendizaje activo.

7. Referencias bibliográficas

- [1] P. Bourque y R. E. Fairley, Eds., *SWEBOK: Guide to the Software Engineering Body of Knowledge*, Version 3.0. Los Alamitos, CA: IEEE Computer Society, 2014. [En línea]. Disponible en: <http://www.swebok.org/>
- [2] K. Beck *et al.*, “Manifesto for Agile Software Development”, *Manifesto for Agile Software Development*. 2001. [En línea]. Disponible en: <http://www.agilemanifesto.org/>
- [3] G. Maturro, C. Fontán, y F. Raschetti, “A Systematic Mapping Study on Soft Skills in Software Engineering”, *Journal of Universal Computer Science*, vol. 25, n° 1, pp. 16–41, 2019.
- [4] G. Maturro y M. Solari, “Soft skills in software engineering in higher education. An initial study of its state in Latin America”, en *XXII Conferencia Iberoamericana de Software Engineering (CIBSE 2019)*, La Habana, Cuba, 2019.
- [5] M. Moore y C. Potts, “Learning by doing: Goals and experiences of two software engineering project courses”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 750 LNCS, pp. 151–164, 1994, doi: 10.1007/BFB0017611/COVER.
- [6] R. Williams, “Education for the Profession Formerly Known as Engineering.”, *Chronicle of Higher Education*. 1/24/2003, Vol. 49 Issue 20, pB12. 2p. Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://www.chronicle.com/article/education-for-the-profession-formerly-known-as-engineering/>
- [7] V. Garousi, G. Giray, E. Tuzun, C. Catal, y M. Felderer, “Closing the Gap Between Software Engineering Education and Industrial Needs”, *IEEE Softw*, vol. 37, n° 2, pp. 68–77, mar. 2020, doi: 10.1109/MS.2018.2880823.

- [8] M. R. Marques, A. Quispe, y S. F. Ochoa, “A systematic mapping study on practical approaches to teaching software engineering”, en *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, IEEE, oct. 2014, pp. 1–8. doi: 10.1109/FIE.2014.7044277.
- [9] M. Prince, “Does Active Learning Work? A Review of the Research”, *Journal of Engineering Education*, vol. 93, n° 3, pp. 223–231, jul. 2004, doi: 10.1002/j.2168-9830.2004.tb00809.x.
- [10] B. Cartaxo *et al.*, “Software Engineering Research Community Viewpoints on Rapid Reviews”, en *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, IEEE, sep. 2019, pp. 1–12. doi: 10.1109/ESEM.2019.8870144.
- [11] R. E. Levine y P. D. Hudes, *How-to Guide for Team-Based Learning*. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-62923-6.
- [12] I. Sommerville, *Software engineering*. Pearson, 2011.
- [13] D. H. Schunk, *Teorías del aprendizaje*. Pearson Educación, 1997.
- [14] R. M. Duncan, “Piaget and Vygotsky Revisited: Dialogue or Assimilation?”, *Developmental Review*, vol. 15, n° 4, pp. 458–472, dic. 1995, doi: 10.1006/drev.1995.1019.
- [15] F. Szabó y I. Csépes, “Constructivism in language pedagogy”, *Hungarian Educational Research Journal*, vol. 13, n° 3, pp. 405–417, nov. 2022, doi: 10.1556/063.2022.00136.
- [16] G. M. Bodner, “Constructivism: A theory of knowledge”, *J Chem Educ*, vol. 63, n° 10, p. 873, oct. 1986, doi: 10.1021/ed063p873.
- [17] B. S. Bloom, *Taxonomy of educational objectives; the classification of educational goals*. 1956.

- [18] Anderson L. W., Sosniak L. A., Bloom B. S., y National Society for the Study of Education, *Bloom's taxonomy : a forty-year retrospective*. University of Chicago Press, 1994.
- [19] A. Amer, "Reflections on Bloom's Revised Taxonomy", *Electronic Journal of Research in Educational Psychology*, vol. 4, n° 1, pp. 213–230, 2006, [En línea]. Disponible en: <http://www.redalyc.org/articulo.oa?id=293123488010>
- [20] D. P. Ausubel, *The psychology of meaningful verbal learning; an introduction to school learning*. Grune & Stratton, 1968.
- [21] T. G. K. Bryce y E. J. Blown, "Ausubel's meaningful learning re-visited", *Current Psychology*, abr. 2023, doi: 10.1007/s12144-023-04440-4.
- [22] M. Rivas Navarro y Comunidad de Madrid, *Procesos cognitivos y aprendizaje significativo*. Comunidad de Madrid. Consejería de Educación. Viceconsejería de Organización Educativa, 2008.
- [23] D. P. Ausubel, "Schemata, Cognitive Structure, and Advance Organizers: A Reply to Anderson, Spiro, and Anderson", *Am Educ Res J*, vol. 17, n° 3, pp. 400–404, may 1980, doi: 10.3102/00028312017003400.
- [24] D. P. Ausubel, *The Acquisition and Retention of Knowledge: A Cognitive View*. Dordrecht: Springer Netherlands, 2000. doi: 10.1007/978-94-015-9454-7.
- [25] D. P. Ausubel, "The use of advance organizers in the learning and retention of meaningful verbal material.", *J Educ Psychol*, vol. 51, n° 5, pp. 267–272, oct. 1960, doi: 10.1037/h0046669.
- [26] S. Vitello, J. Greatorex, y S. Shaw, "What is competence? A shared interpretation of competence to support teaching, learning and assessment Research Report", 2021. [En línea]. Disponible en: <https://www.cambridge.org/>
- [27] L. Jabif, *La Docencia Universitaria Bajo un Enfoque de Competencias*. Valdivia: Universidad Austral de Chile, 2007.

- [28] K. Umbleja, V. Kukk, M. Jaanus, y A. Udal, “New Concepts of Automatic Answer Evaluation in Competence Based Learning”, *IEEE Global Engineering Education Conference (EDUCON)*, 2014.
- [29] A. Sánchez Villa y M. Poblete Ruiz, *Competence-based learning A proposal for the assessment of generic competences*. Bilbao: University of Deusto, 2008.
- [30] S. Damek, H. Söbke, F. Weise, y M. Reichelt, “Teaching (Meta) Competences for Digital Practice Exemplified by Building Information Modeling Work Processes”, *Knowledge*, vol. 2, n° 3, pp. 452–464, sep. 2022, doi: 10.3390/knowledge2030027.
- [31] R. B. Brown, “Refrain the Competency Debate”, *Manag Learn*, vol. 25, n° 2, pp. 289–299, jun. 1994, doi: 10.1177/1350507694252008.
- [32] M. L. Matteson, L. Anderson, y C. Boyden, “‘Soft Skills’: A Phrase in Search of Meaning”, *portal: Libraries and the Academy*, vol. 16, n° 1, pp. 71–88, 2016, doi: 10.1353/pla.2016.0009.
- [33] E. Dell’Aquila, Davide Marocco, M. Ponticorvo, A. Di Ferdinando, M. Schembri, y O. Miglino, *Educational Games for Soft-Skills Training in Digital Environments*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-06311-9.
- [34] G. G. Borges y R. C. Gratão de Souza, “Skills development for software engineers: Systematic literature review”, *Inf Softw Technol*, vol. 168, p. 107395, abr. 2024, doi: 10.1016/j.infsof.2023.107395.
- [35] S. P. Guerra-Báez, “A panoramic review of soft skills training in university students”, *Psicología Escolar e Educativa*, vol. 23, 2019, doi: 10.1590/2175-35392019016464.
- [36] S. Rajuli y N. Baharun, “A Comparative Study of Problem-Based and Traditional Teaching in Computing Subjects”, en *Proceedings of the International Conference on Computing, Mathematics and Statistics (iCMS 2015)*, Singapore: Springer Singapore, 2017, pp. 275–284. doi: 10.1007/978-981-10-2772-7_28.

- [37] L. Tan, “Comparison of PBL and the Traditional Teaching Method in the Teaching of Economics”, 2011, pp. 567–572. doi: 10.1007/978-3-642-23753-9_91.
- [38] J. K. Crosier, S. V. G. Cobb, y J. R. Wilson, “Experimental Comparison of Virtual Reality with Traditional Teaching Methods for Teaching Radioactivity”, *Educ Inf Technol (Dordr)*, vol. 5, n° 4, pp. 329–343, 2000, doi: 10.1023/A:1012009725532.
- [39] A. W. Chickering y Z. F. Gamson, “Appendix A: Seven principles for good practice in undergraduate education”, *New Directions for Teaching and Learning*, vol. 1991, n° 47, pp. 63–69, sep. 1991, doi: 10.1002/tl.37219914708.
- [40] C. C. Bonwell y J. A. Eison, “Active Learning: Creating Excitement in the Classroom”, 1991. [En línea]. Disponible en: www.eric.ed.gov
- [41] L. Springer, M. E. Stanne, y S. S. Donovan, “Effects of Small-Group Learning on Undergraduates in Science, Mathematics, Engineering, and Technology: A Meta-Analysis”, *Rev Educ Res*, vol. 69, n° 1, pp. 21–51, mar. 1999, doi: 10.3102/00346543069001021.
- [42] M. J. Prince y R. M. Felder, “Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases”, *Journal of Engineering Education*, vol. 95, n° 2, pp. 123–138, abr. 2006, doi: 10.1002/j.2168-9830.2006.tb00884.x.
- [43] L. K. Michaelsen y M. Sweet, “The essential elements of team-based learning”, *New Directions for Teaching and Learning*, vol. 2008, n° 116, pp. 7–27, dic. 2008, doi: 10.1002/tl.330.
- [44] Team-Based Learning Collaborative, “Team-Based Learning FAQ”. Accedido: 9 de enero de 2024. [En línea]. Disponible en: https://www.teambasedlearning.org/answers-to-faqs/#q3_2

- [45] P. Ruder, M. H. Maier, y S. P. Simkins, “Getting started with team-based learning (TBL): An introduction”, *J Econ Educ*, vol. 52, n° 3, pp. 220–230, jun. 2021, doi: 10.1080/00220485.2021.1925187.
- [46] Iowa State University Center for Excellence in Learning and Teaching, “226 Active Learning Techniques”. Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://www.celt.iastate.edu/wp-content/uploads/2017/03/CELT226activelearningtechniques.pdf>
- [47] Cornell University Center for Teaching Innovation, “Getting started with active learning techniques”. Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://teaching.cornell.edu/getting-started-active-learning-techniques>
- [48] University of Minnesota Center for Educational Innovation, “Active Learning”. Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://cei.umn.edu/teaching-resources/active-learning>
- [49] M. Daun, A. M. Grubb, V. Stenkova, y B. Tenbergen, “A systematic literature review of requirements engineering education”, *Requir Eng*, vol. 28, n° 2, pp. 145–175, jun. 2023, doi: 10.1007/s00766-022-00381-9.
- [50] H. Kondo y A. Hazeyama, “Systematic Literature Review on Educational Effectiveness of Project-Based Learning for Software Development”, en *2022 29th Asia-Pacific Software Engineering Conference (APSEC)*, IEEE, dic. 2022, pp. 584–585. doi: 10.1109/APSEC57359.2022.00092.
- [51] M. R. de A. Souza, L. Veado, R. T. Moreira, E. Figueiredo, y H. Costa, “A systematic mapping study on game-related methods for software engineering education”, *Inf Softw Technol*, vol. 95, pp. 201–218, mar. 2018, doi: 10.1016/j.infsof.2017.09.014.
- [52] S. Khangura, K. Konnyu, R. Cushman, J. Grimshaw, y D. Moher, “Evidence summaries: The evolution of a rapid review approach”, *Syst Rev*, vol. 1, n° 1, feb. 2012, doi: 10.1186/2046-4053-1-10.

- [53] S. Rico, “Guidelines for conducting interactive rapid reviews in software engineering-from a focus on technology transfer to knowledge exchange”, 2020. [En línea]. Disponible en: <http://serg.cs.lth.se/>
- [54] A. Paez, “Gray literature: An important resource in systematic reviews”, *J Evid Based Med*, vol. 10, n° 3, pp. 233–240, ago. 2017, doi: 10.1111/jebm.12266.
- [55] University of Oxford Centre for Teaching and Learning, “Encouraging active learning in large class teaching”. Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://wwwctl.ox.ac.uk/large-class-teaching>
- [56] Harvard University The Derek Bok Center for Teaching and Learning, “Active Learning”. Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://bokcenter.harvard.edu/active-learning>
- [57] Massachusetts Institute of Technology, “Active Learning”. Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://web.mit.edu/edtech/themes/activelearning.html>
- [58] Centro de Actualización en la Enseñanza Superior, “Métodos y técnicas de enseñanza”. Accedido: 9 de enero de 2024. [En línea]. Disponible en: <https://caes.ort.edu.uy/herramientas-para-la-docencia/metodos-y-tecnicas-de-ensenanza>
- [59] J. A. Hurtado, A. C. Useche, y B. S. Masiero, “Project-Based Learning: Authentic Engineering Assessment Supported by Model Design”, *International Journal of Engineering Pedagogy (iJEP)*, vol. 13, n° 6, pp. 17–32, sep. 2023, doi: 10.3991/ijep.v13i6.38539.
- [60] S. Williams van Rooij, “Scaffolding project-based learning with the project management body of knowledge (PMBOK®)”, *Comput Educ*, vol. 52, n° 1, pp. 210–219, ene. 2009, doi: 10.1016/j.compedu.2008.07.012.
- [61] H. Yun, S. Kim, y E.-R. Han, “Latent profile analysis on the effectiveness of tutor performance: Influence on medical students’ engagement in blended

- problem-based learning”, *PLoS One*, vol. 18, n° 10, p. e0292843, oct. 2023, doi: 10.1371/JOURNAL.PONE.0292843.
- [62] S. A. Nicolaou, A. Heraclides, C. S. Constantinou, S. Loizou, y D. J. Gillott, “One size doesn’t fit all: PBL tutor training and development”, *Interdisciplinary Journal of Problem-based Learning*, vol. 15, n° 2, dic. 2021, doi: 10.14434/IJPBL.V15I2.30267.
- [63] V. Duarte, M. Cleveland-Slimming, C. Vidal, y S. Contreras, “Information System Development by Using Agile Teamwork and Service-Learning”, *IEEE Transactions on Education*, vol. 66, n° 5, pp. 431–441, oct. 2023, doi: 10.1109/TE.2023.3310243.
- [64] L. M. Shea, D. Harkins, S. Ray, y L. I. Grenier, “How Critical is Service-Learning Implementation?”, *Journal of Experiential Education*, vol. 46, n° 2, pp. 197–214, jun. 2023, doi: 10.1177/10538259221122738.
- [65] S. R. Bandi, G. Joshi, A. Shettar, y R. Kandakatla, “A Systematic Literature Review on Faculty Learning in Service-Learning”, en *2023 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, may 2023, pp. 1–10. doi: 10.1109/EDUCON54358.2023.10125223.
- [66] D. A. Kolb, *Experiential Learning Experience as the Source of Learning and Development*. Prentice-Hall, 1984.
- [67] A. Shore y T. Dinning, “Developing student’s skills and work readiness: an experiential learning framework”, *Journal of Work-Applied Management*, vol. 15, n° 2, pp. 188–199, sep. 2023, doi: 10.1108/JWAM-02-2023-0016.
- [68] D. C. Brandenburg y A. D. Ellinger, “The Future: Just-in-Time Learning Expectations and Potential Implications for Human Resource Development”, *Adv Dev Hum Resour*, vol. 5, n° 3, pp. 308–320, ago. 2003, doi: 10.1177/1523422303254629.

- [69] X. Mao, Y. Lu, L. Yin, T. Wang, y G. Yin, “Model and Practice of Crowd-Based Education”, en *Web and Big Data*, 2018, pp. 293–305. doi: 10.1007/978-3-030-01298-4_25.
- [70] M. Jaanus, K. Umbleja, A. Udal, y K. Parnamets, “Integrated labs for electrical engineering courses in competence based learning environment-practical experience”, *2019 Electric Power Quality and Supply Reliability Conference and 2019 Symposium on Electrical Engineering and Mechatronics, PQ and SEEM 2019*, jun. 2019, doi: 10.1109/PQ.2019.8818266.
- [71] A. Alammery, “Blended learning models for introductory programming courses: A systematic review”, *PLoS One*, vol. 14, n° 9, p. e0221765, sep. 2019, doi: 10.1371/journal.pone.0221765.
- [72] A. Manasrah, M. Masoud, Y. Jaradat, M. Irshaidat, N. A. Shaban, y A. Zerek, “Students Engagement in Blended Learning: Evidence from Moodle A Case Study from Engineering Courses”, en *Proceeding - 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering, MI-STA 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 435–439. doi: 10.1109/MI-STA57575.2023.10169635.
- [73] A. M. O’Donnell y A. King, *Cognitive perspectives on peer learning*. en The Rutgers Invitational Symposium On Education Series. Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers, 1999.
- [74] M. Laal y M. Laal, “Collaborative learning: what is it?”, *Procedia Soc Behav Sci*, vol. 31, pp. 491–495, 2012, doi: 10.1016/j.sbspro.2011.12.092.
- [75] M. Videnovik, T. Vold, L. Kiønig, A. Madevska Bogdanova, y V. Trajkovik, “Game-based learning in computer science education: a scoping literature review”, *Int J STEM Educ*, vol. 10, n° 1, p. 54, sep. 2023, doi: 10.1186/s40594-023-00447-2.

- [76] C. Paez-Quinde, A. Chasipanta-Nieves, C. A. Hernandez-Davila, y J. Arevalo-Peralta, “Flipped classroom in the meaningful learning of the students of the Basic Education Career: Case study Technical University of Ambato”, en *2022 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, mar. 2022, pp. 785–789. doi: 10.1109/EDUCON52537.2022.9766792.
- [77] Y. Dong, H. Yin, S. Du, y A. Wang, “The effects of flipped classroom characterized by situational and collaborative learning in a community nursing course: A quasi-experimental design”, *Nurse Educ Today*, vol. 105, p. 105037, oct. 2021, doi: 10.1016/j.nedt.2021.105037.
- [78] Y. Choi y E. Kim, “A case study on the evaluation of discussion and debate learning effectiveness in a dental hygiene ethics class”, *European Journal of Dental Education*, vol. 26, n° 2, pp. 223–231, may 2022, doi: 10.1111/eje.12690.
- [79] Krismadinata, U. I. Kurnia, R. Mulya, y U. Verawardina, “The Interactive Multimedia Learning for Power Electronics Course”, *International journal of online and biomedical engineering*, vol. 18, n° 7, pp. 44–56, 2022, doi: 10.3991/ijoe.v18i07.30029.
- [80] M. Galster, A. Mitrovic, y M. Gordon, “Toward Enhancing the Training of Software Engineering Students and Professionals Using Active Video Watching”, *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2018.
- [81] G. Siemens, “Connectivism: A Learning Theory for the Digital Age”, 2004.
- [82] Brown University, “Interactive Classroom Activities”, <https://www.brown.edu/sheridan/teaching-learning-resources/teaching-resources/classroom-practices/active-learning/interactive>. Accedido: 3 de febrero de 2024. [En línea]. Disponible en: <https://www.brown.edu/sheridan/teaching-learning-resources/teaching-resources/classroom-practices/active-learning/interactive>

- [83] Carnegie Mellon University, “Using Concept Tests”, <https://www.cmu.edu/teaching/assessment/assesslearning/concepTests.html>.
Accedido: 3 de febrero de 2024. [En línea]. Disponible en:
<https://www.cmu.edu/teaching/assessment/assesslearning/concepTests.html>
- [84] V. Lian, E. Varoy, y N. Giacaman, “Learning Object-Oriented Programming Concepts Through Visual Analogies”, *IEEE Transactions on Learning Technologies*, vol. 15, n° 1, pp. 78–92, feb. 2022, doi: 10.1109/TLT.2022.3154805.
- [85] L. Kong, “Empirical Study on the Effects of the Application of Virtual Reality to Experiential Education on Students’ Learning Attitude and Learning Effectiveness”, *Revista de Cercetare si Interventie Sociala*, vol. 73, pp. 288–298, jun. 2021, doi: 10.33788/rcis.73.18.
- [86] Universidad ORT Uruguay, “Documento 220”. Universidad ORT Uruguay, Montevideo, febrero de 2022.
- [87] R. L. Bartlett, “A Flip of the Coin. A Roll of the Die: An Answer to the Free-Rider Problem in Economic Instruction”, 1995.

ANEXO 1

Guion entrevista semi-estructurada a docente de la Cátedra de Ingeniería de Software:

Q1. ¿Qué materias has dictado?

Q2. ¿Cómo arrancó/arrancaste TBL?

Q3. ¿Cómo fue la primera vez en TBL?

Q4. ¿Alguna dificultad que te haya marcado?

Q5. ¿Qué percibiste de un semestre a otro aplicando TBL?

Q6. ¿Cuáles son los principales cambios que recuerdas sobre la aplicación de TBL?

Q7. ¿Cómo se resolvió que lleva y qué no lleva puntaje?

Q8. ¿Qué no hacemos y querríamos hacer?

Q9. ¿Cómo vives los ejercicios de aplicación de TBL?

Q10. ¿Cuál es tu valoración global sobre las actividades de enseñanza que realizamos en nuestras asignaturas?

ANEXO 2

