

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>



Artículo original

## Modelos, simuladores e Ingeniería del Software

Models, simulators and Software Engineering

Modelos, simuladores e Engenharia de Software

Sonia Itati Mariño<sup>1</sup>



<https://orcid.org/0000-0003-3529-7003>

Pedro Luis Alfonzo<sup>1</sup>



<https://orcid.org/0000-0001-5447-8518>

<sup>1</sup>Universidad Nacional del Nordeste. Argentina.



[simarinio@yahoo.com](mailto:simarinio@yahoo.com);  
[plalfonzo@hotmail.com](mailto:plalfonzo@hotmail.com)

**Recibido:** 14 de julio, 2023

**Aceptado:** 6 de diciembre, 2024

### RESUMEN

Se presenta un enfoque interdisciplinario que relaciona los modelos, los simuladores y la Ingeniería del Software. En particular desde la Ingeniería del Software se abordan y adaptan cuestiones de agilidad y la guía de conocimiento SWEBOK. La metodología, para el logro de la propuesta trata la exploración de la literatura y una adaptación de SCRUM para asistir en la gestión de proyectos educativos y contribuir al

proceso de enseñanza-aprendizaje. Además, se explicita las prácticas desarrolladas en una asignatura de grado con la finalidad de lograr aprendizajes significativos, centrados en los estudiantes orientados al aprendizaje basado en proyectos. Se concluye que la guía SWEBOK proporciona áreas de conocimiento que son tratadas desde la modelización y simulación de sistemas, contribuyendo con aprendizajes significativos a través de la apropiación e integración de conocimientos disciplinares. A futuro, se identificarán y analizarán otras áreas de conocimiento de la Ingeniería del Software que se vincularían con el abordaje propuesto en la asignatura centrándose en el perfil de los estudiantes.

**Palabras clave:** SCRUM; educación superior; modelos; simuladores; Ingeniería del Software.

### ABSTRACT

An interdisciplinary approach is presented, linking models, simulators and software engineering. In particular from the Software Engineering are addressed and adapted agility issues and the SWEBOK knowledge guide. The methodology, for the achievement of the proposal deals with the exploration of literature and an adaptation of SCRUM to assist in the management of educational projects and contribute to the teaching-learning process. In addition, the practices developed in a degree subject are explicit with the aim of achieving meaningful learning, focused on students oriented to project-based learning. It is concluded that the SWEBOK guide provides areas of knowledge that are addressed from the modelling and simulation of systems, contributing to significant learning through the appropriation and integration of disciplinary knowledge. In the future, other areas of knowledge in Software Engineering that would be linked to the proposed approach in the subject will be identified and analyzed focusing on the profile of students.

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

**Keywords:** SCRUM; higher education; models; simulators; Software Engineering.

## RESUMO

É apresentada uma abordagem interdisciplinar que relaciona modelos, simuladores e Engenharia de Software. Principalmente na Engenharia de Software, questões de agilidade e o guia de conhecimento SWEBOK são abordados e adaptados. A metodologia, para concretização da proposta, trata da exploração de literatura e de uma adaptação do SCRUM para auxiliar na gestão de projetos educacionais e contribuir no processo de ensino-aprendizagem. Além disso, são explicadas as práticas desenvolvidas em uma disciplina de graduação com o objetivo de alcançar uma aprendizagem significativa, voltada para alunos orientados para a aprendizagem baseada em projetos. Conclui-se que o guia SWEBOK disponibiliza áreas de conhecimento que são tratadas a partir da modelagem e simulação de sistemas, contribuindo com uma aprendizagem significativa através da apropriação e integração de conhecimentos disciplinares. Futuramente serão identificadas e analisadas outras áreas do conhecimento da Engenharia de Software que estariam vinculadas à abordagem proposta na disciplina, com foco no perfil dos alunos.

**Palavras-chave:** SCRUM; ensino superior; modelos; simuladores; Engenharia de Software.

## INTRODUCCIÓN

La complejidad y la emergencia de diversos fenómenos del mundo actual evidencian la necesidad de modelar abstracciones de la realidad para su posterior experimentación mediante simuladores de software. Los modelos de simulación se utilizan en una amplia variedad de contextos, como los negocios, el comercio, la industria, las ingenierías, la producción, el

transporte y la academia, entre otros. La construcción de estos simuladores, representativos de los modelos ideados, recurre a áreas de conocimiento de la Ingeniería del Software (IS), una de las disciplinas de la Informática.

## La Ingeniería del Software y la Guía SWEBOK

La Ingeniería del Software es una disciplina ingenieril orientada a la producción de software. Comprende todas las etapas del proceso, desde las primeras fases de especificación del sistema hasta su mantenimiento una vez que entra en uso. Asociaciones profesionales como la IEEE Computer Society y la Association for Computing Machinery unieron esfuerzos para desarrollar la guía del Cuerpo de Conocimientos de la Ingeniería de Software, conocida como SWEBOK (Software Engineering Body of Knowledge). Este documento se considera una guía fundamental de la disciplina. Entre sus características, destacadas por Bourque y Fairley (2014), se incluyen:

- Caracterizar los contenidos de la Ingeniería del Software.
- Proveer acceso, a través de diversas temáticas, al conjunto de conocimientos de la Ingeniería del Software.
- Promover una visión consistente de la Ingeniería del Software a nivel mundial.
- Clarificar la relación de la Ingeniería del Software con otras disciplinas, como las Ciencias de la Computación o las Matemáticas.
- Proveer una base para el desarrollo curricular y la creación de materiales de certificación.

En el trabajo de Bourque y Fairley (2014) se definen 15 áreas de conocimiento, que son: requisitos de software, diseño de software, construcción de software, pruebas de software, mantenimiento de software, gestión de la configuración de software, gestión de ingeniería de software, proceso de ingeniería de software, modelos y métodos de ingeniería de software,

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

calidad del software, práctica profesional de la ingeniería de software, economía de la ingeniería de software, fundamentos de computación, fundamentos matemáticos y fundamentos de ingeniería.

Otros trabajos que abordan la guía SWEBOK en el ámbito de la educación superior pueden encontrarse en Alenezi y Akour (2022), Abbran et al. (2022) y Kamthan y Washizaki (2022).

### La gestión ágil de proyectos

La Industria del Software, para responder con eficiencia y eficacia, debe adoptar agilidad en sus proyectos. En el informe 15th Annual State of Agile, se destacó a SCRUM como el enfoque ágil más popular, con un 66 % de las personas encuestadas identificándolo como su preferido. Adicionalmente, un 15 % mencionó derivaciones de SCRUM, como ScrumBan (9 %) y Scrum/XP (6 %) (Hernández Sola, 2021).

En contextos de educación superior, existe una tendencia creciente a adoptar y adaptar metodologías ágiles. Entre los aportes más relevantes se encuentran los trabajos de Vila Grau y Capuz Rizo (2021), Capuñay Uceda y Antón Pérez (2021), Arias, Tomaselli y Tortosa (2021), y Ruiz de Mendarozqueta, Bustos y Colla (2022).

Una síntesis de estas contribuciones evidencia la diversidad de enfoques en los que SCRUM, como metodología ágil, se aborda desde revisiones de la literatura, en los procesos de enseñanza y aprendizaje, y como apoyo en la gestión de proyectos educativos. Además, el análisis reflexivo en este ámbito contribuye tanto teórica como metodológicamente a esta propuesta.

### Contexto

La asignatura optativa "Modelos y Simulación" pertenece al Área de Ciencias y Métodos Computacionales de la carrera Licenciatura en Sistemas de Información.

En esta asignatura, se desarrolla un proyecto orientado a la articulación de actividades de docencia e investigación aplicada, en concordancia con algunas de las misiones de la Universidad. En particular, los contenidos se enfocan en la resolución de abstracciones de problemas del mundo real, con el objetivo de identificar y proponer alternativas de solución. Asimismo, se promueve la identificación y sistematización de experiencias vinculadas a proyectos de I+D+i.

Con el propósito de contribuir a proyectos de extensión e investigación aplicada, transfiriendo los conocimientos docentes y específicos de las técnicas de modelado y simulación, se identifican problemas reales que se abstraen y resuelven utilizando estas herramientas. Además, debido a la ubicación de la asignatura en el plan de estudios, los estudiantes implicados pueden desempeñarse como pasantes, becarios o trabajar en diversos contextos.

Cabe destacar que, en las modelizaciones realizadas, se incentiva la generación de numerosos experimentos, su tabulación, representación gráfica y posterior análisis. Esto fomenta habilidades como el análisis de resultados y la argumentación, esenciales para la formación profesional de los alumnos.

Actividades destacadas de la asignatura:

- Problematización y abstracción: Situaciones reales factibles de modelizar y simular se abordan a partir de problemáticas planteadas en la asignatura.
- Identificación de métodos y herramientas:
- Selección, estudio y evaluación de métodos para modelizar problemas propios de la simulación.
- Identificación y evaluación de herramientas de software para la generación de simuladores y modelos abstractos de problemas reales. Estas herramientas son seleccionadas por los estudiantes, y los simuladores se utilizan

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

como artefactos para ejecutar experimentos, analizar resultados e inferir soluciones alternativas. A futuro, se podrían proponer estudios comparativos.

- Construcción de simuladores: Desarrollo de artefactos de software que permitan la experimentación.
- Verificación: Corroboración de la correcta implementación con respecto al modelo conceptual, es decir, la correspondencia con las especificaciones y los supuestos aceptables según el propósito del modelo y del simulador.
- Experimentación: Evaluación del desempeño de los algoritmos de simulación estudiados, validando el uso de diversos lenguajes de programación.
- Articulación de conocimientos: Integración de temas como probabilidades, programación y gestión de proyectos. Se fomenta la elección de temas tratados en asignaturas previas, a los cuales se propone incorporar el módulo de simulación.

En esta asignatura, los estudiantes, de forma individual o en equipos de hasta 2 o 3 integrantes, deben desarrollar un trabajo integrador que sintetice, mediante una metodología propia de la simulación de sistemas, una abstracción de un caso de estudio. Estos simuladores aplican mayormente modelos de inventarios o colas de espera, y se relacionan con la metodología activa conocida como aprendizaje basado en proyectos (Botella Nicolás y Ramos Ramos, 2020; Villanueva Morales, Ortega Sánchez y Díaz Sepúlveda, 2022; Soriano, Rosero, Guzmán y Nieves, 2024).

El artículo tiene como objetivo presentar un enfoque interdisciplinario que relaciona los modelos, los simuladores y la Ingeniería del Software.

## MATERIALES Y MÉTODOS

El enfoque metodológico para el logro de los resultados comprendió las siguientes fases.

Fase 1. Revisión exploratoria de la literatura:

Se focalizó en la selección y análisis documental de artículos relacionados con los métodos ágiles y la guía de conocimiento SWEBOK. Siguiendo a Romillo y Polaino (2019, p. 4), esta fase "posibilitó identificar, mediante la utilización del método de análisis y síntesis, los principales elementos y su interrelación, que resultaban determinantes para la toma de decisiones" en relación con la presente investigación y desarrollo.

Fase 2. Definición y delimitación del contexto en que se sitúa la I+D:

Se estableció como contexto de la investigación aplicada la asignatura Modelos y Simulación de una carrera de grado en sistemas, articulando con algunas líneas de proyectos de I+D acreditados por la universidad.

Fase 3. Adecuación de la metodología ágil SCRUM:

Su elección se fundamentó en el diseño de una estrategia para sistematizar la producción de conocimiento empírico sustentado en la teoría.

Fase 4. Identificación de áreas de conocimiento SWEBOK implicadas en el desarrollo:

En el diseño y desarrollo de la investigación se aplicaron métodos del nivel teórico, como el analítico-sintético, en este caso para identificar antecedentes en torno a áreas de la Ingeniería del Software y determinar cuáles se articulan con las actividades desarrolladas en la asignatura, específicamente la construcción de simuladores software.

Se optó por el método inductivo-deductivo, con la finalidad de avanzar desde el conocimiento

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

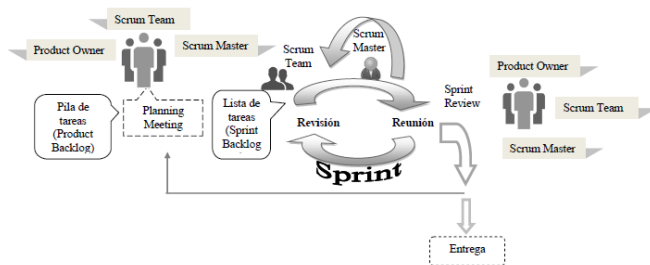
<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

general al particular, especialmente para analizar y reflexionar en torno a la vinculación entre algunas áreas de conocimiento de la guía SWEBOK y el diseño y desarrollo de simuladores de modelos abstraídos de la realidad para resolver problemas identificados desde la academia.

A nivel empírico, sustentado en los métodos teóricos, se utilizó el análisis de observaciones y notas realizadas durante el periodo 2021, en un contexto de aislamiento social preventivo y obligatorio (ASPO). Durante este periodo, se abordaron diversas situaciones según las normativas institucionales vigentes. Los simuladores construidos en el marco de la asignatura se constituyeron en una fuente de información clave para sustentar la propuesta.

## RESULTADOS

A continuación, se describe la adecuación de la metodología ágil SCRUM. La figura 1 presenta roles, artefactos y prácticas de SCRUM incluidos en la propuesta.



**Fig. 1-** Roles, artefactos y practicas SCRUM

### Definición de roles

En un contexto de educación superior y adaptando Scrum, se establecieron los siguientes roles:

### Roles Scrum

**Product Owner:** Es el dueño del producto, representado por el profesor responsable y los docentes de la asignatura. Define y prioriza la lista de tareas en función de los requerimientos del proyecto y la planificación ágil para lograr los aprendizajes establecidos.

**Scrum Master:** Es el líder del proyecto. Asegura el cumplimiento de las actividades y resuelve problemas que surjan para alcanzar los objetivos de enseñanza-aprendizaje definidos en la planificación. Este rol es asumido por los docentes, quienes guían la ejecución de las tareas.

**Scrum Team:** Es el equipo responsable de ejecutar la propuesta, compuesto por los estudiantes, bajo la guía del Scrum Master.

### Prácticas y artefactos Scrum

**Product Backlog:** Representa los requerimientos de cada proyecto en una lista de tareas asociadas al modelo y su simulador.

**Gestión de riesgos:** Se lleva a cabo continuamente mediante reuniones programadas en cada clase, Planning Meetings y Sprint Reviews, orientadas al monitoreo, seguimiento y retroalimentación según la planificación.

**Planning Meeting:** Reunión de planificación del Sprint a partir del Product Backlog. Participan el Product Owner, el Scrum Master y el Scrum Team. Estas reuniones se concretan durante las clases previstas en el cronograma y pueden complementarse con tutorías o consultas específicas.

**Sprint Backlog:** Contiene las tareas seleccionadas del Product Backlog que se abordarán en cada Sprint.

**Gestión del Sprint Backlog:** Al inicio de cada iteración, se seleccionan los requerimientos y se estima el esfuerzo de cada tarea. Estas

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

iteraciones se relacionan con guías de trabajos prácticos y de laboratorio que, de manera incremental, contribuyen al desarrollo del Trabajo Final Integrador (TFI).

**Sprint:** Cada Sprint tiene una duración estimada de dos semanas y produce un módulo funcional del trabajo final. Por ejemplo, construir una muestra artificial, un simulador de teoría de colas o un simulador de inventarios.

**Reuniones:** Se realizan de manera presencial, virtual o asincrónica para garantizar el logro de los objetivos. La participación activa de todos los integrantes del equipo es esencial para la retroalimentación y mejora continua.

**Sprint Review:** Al finalizar cada Sprint, el Scrum Team presenta los avances mediante exposiciones orales, demostraciones de entregables, informes u otros recursos. Participan el Product Owner, el Scrum Master y el Scrum Team.

**Gráficos de Burn-down:** Permiten visualizar y analizar el progreso del equipo, proporcionando una representación gráfica de las tareas completadas y pendientes. Esto asegura que las entregas se realicen dentro de los plazos establecidos, lo cual incide en la promoción o aprobación de la asignatura.

### **Sprints contemplados en la propuesta**

**Sprint 1:** Delimitación de la propuesta, diseñando un caso de estudio adaptado a la estrategia metodológica expuesta en la subsección "Contexto".

**Sprint 2:** Análisis de las guías de trabajos prácticos y de laboratorio propuestas por la asignatura, así como de las producciones estudiantiles del ciclo lectivo 2021.

**Sprint 3:** Profundización en las áreas de conocimiento de la guía SWEBOK. Este análisis permitió identificar conexiones entre la Ingeniería del Software y la construcción de software para experimentación y simulación,

conformando un portafolio que sustenta el Trabajo Final Integrador.

**Sprint 4:** Determinación de las áreas de conocimiento de la guía SWEBOK representadas en los trabajos prácticos y de laboratorio. Los simuladores desarrollados incluyen tópicos relacionados con "Fundamentos matemáticos", "Construcción de Software" y "Práctica Profesional de la Ingeniería del Software".

## **DISCUSIÓN**

El área de conocimiento de Fundamentos Matemáticos (Bourque y Fairley, 2014) ayuda a los ingenieros de software a comprender la lógica, la cual se traduce en código en lenguaje de programación. La lógica y el razonamiento son esenciales en las matemáticas que un ingeniero de software debe dominar. Además, se abordan técnicas que permiten representar y llevar a cabo el razonamiento y el juicio de manera precisa y, por lo tanto, matemática. Se considera que solo es posible escribir un programa de computación para un problema determinado si se sigue una lógica coherente. Así, el objetivo de esta área de conocimiento es facilitar el desarrollo de habilidades para identificar y describir dicha lógica.

El término "desarrollo de software" hace referencia a la creación detallada del software y al área de conocimiento denominada Construcción de Software, según la guía de Bourque y Fairley (2014). Esta guía presenta una descripción detallada sobre cómo elaborarlo. Expone que la construcción de software es una actividad en la que el ingeniero de software debe enfrentar las limitaciones del mundo real, que es caótico y cambiante, y debe hacerlo con precisión. Debido a la influencia de estas restricciones del mundo real, la construcción de software está más impulsada por consideraciones prácticas que otras áreas de conocimiento. Los lenguajes de desarrollo incluyen todas las formas de comunicación mediante las cuales una persona puede

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

especificar una solución ejecutable para un determinado problema. En esta asignatura, se promueve la construcción de simuladores de software, utilizando las herramientas previamente estudiadas y, preferentemente, las de los propios estudiantes.

En cuanto al área de conocimiento "Práctica Profesional de la Ingeniería del Software" (Bourque y Fairley, 2014), se ocupa del conocimiento, las habilidades y las actitudes que los ingenieros deben poseer para practicar la ingeniería de software de manera profesional, responsable y ética. Dado que los productos de software tienen aplicaciones generalizadas en la vida social y personal, su calidad puede tener un profundo impacto en nuestro bienestar personal y social. En particular, el desarrollo de simuladores es una estrategia de apoyo a la toma de decisiones abstraídas del contexto, que permite inferir posibles resultados sin intervenir directamente con el entorno. Por ello, se considera que actividades como esta contribuyen a articular la interacción entre la universidad, la empresa y el contexto.

Aunque en los estudios de Abran et al. (2022), Alenezi y Akour (2022), y Kamthan y Washizaki (2022) se menciona el uso de la guía SWEBOK en cuestiones curriculares, sus enfoques no están relacionados con una asignatura de matemáticas aplicadas dentro del currículo de la disciplina de Ciencias de la Computación. Este hecho sustenta el enfoque diferenciador de la presente propuesta. Un área de conocimiento de la guía SWEBOK es el testing. Alenezi y Akour (2022) proponen un curso que contempla los criterios y normas de SWEBOK y ABET con el fin de coincidir con temas especializados de la Ingeniería de Software y contribuir a la formación de las habilidades requeridas por la industria. En la asignatura propuesta, el testing es una actividad relevante, aunque no se ha explicitado en esta propuesta, podría ser objeto de desarrollo en cursos futuros.

Kamthan y Washizaki (2022) sintetizan las ponencias presentadas en la Conferencia Internacional sobre Ingeniería de Software e

Ingeniería del Conocimiento (SEKE, 2022) en torno a la guía SWEBOK. El valor de estas ponencias reside en su capacidad para propiciar la integración de la guía en organizaciones intensivas en software e instituciones educativas. En este sentido, aunque esta propuesta se limita a un curso optativo de grado, se considera que contribuye a la toma de conciencia sobre la necesidad de enfoques interdisciplinarios, especialmente en asignaturas que aportan métodos alternativos para apoyar la toma de decisiones, como las técnicas de simulación de sistemas que abordan abstracciones de la realidad.

En el siglo XXI, se valoran positivamente los enfoques interdisciplinarios en los que el conocimiento, las experiencias y los saberes de diversas disciplinas contribuyen significativamente a la formación. Por ello, las abstracciones de problemas modelados y simulados, al integrar temas de Ingeniería de Software, fortalecen la vinculación entre los ámbitos académicos y profesionales, ampliamente centrados en la Industria del Software.

Siguiendo lo expuesto por Vila Grau y Capuz Rizo (2021), los hallazgos de este artículo contribuyen con evidencia generada en espacios de educación superior a la denominada "expansión de la Agilidad y Scrum en las aulas" (p. 2276). Este es otro ejemplo de cómo abordar de manera interdisciplinaria cuestiones de gestión de proyectos ágiles en contextos educativos.

Otro ejemplo de enfoque interdisciplinario es la articulación desde una asignatura optativa en el área de Ciencias y Métodos Computacionales de una carrera de sistemas, y sus relaciones con áreas de conocimiento de la Ingeniería del Software tratadas en la guía SWEBOK. Específicamente, se trabajó con el aprendizaje basado en problemas (ABP), mencionando ejemplos de lo tratado en Botella Nicolás y Ramos Ramos (2020). Además, Villanueva Morales, Ortega Sánchez y Díaz Sepúlveda (2022) señalan que el ABP implica el trabajo

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

colaborativo, una modalidad que se vislumbra en la agilidad.

Desde un enfoque didáctico-pedagógico, el ABP asegura la apropiación e integración de conocimientos disciplinares, evidenciando nuevas alternativas para generar conocimiento orientado a la toma de decisiones. Por lo tanto, se considera que articular los aspectos de la disciplina en el ABP fortalece los aprendizajes significativos de los estudiantes que se insertan en un mundo laboral complejo. Es decir, se trata de un aprendizaje activo (Soriano et al., 2024) que, a partir de la abstracción de problemas reales, genera experiencias que enriquecen el aprendizaje real y significativo. Coincidiendo con Soriano et al. (2024), se afirma que los estudiantes refuerzan tanto sus conocimientos específicos como sus competencias profesionales.

Como trabajos futuros, se propone explorar otras vinculaciones siguiendo lo expuesto en el programa de la asignatura. De este modo, se visualizarán y explicitarán los estudios y hallazgos de trabajos que integran asignaturas del área, así como de otras áreas de conocimiento de la carrera, como programación, sistemas de información, entre otras.

Desde enfoques centrados en los estudiantes para lograr aprendizajes significativos, se diseñará una encuesta para relevar la percepción sobre las áreas de conocimiento de la Ingeniería del Software, centrada en el perfil de los participantes. Además, como parte de un ciclo de mejora continua, se propone desarrollar un instrumento para sistematizar y medir la motivación de los estudiantes, ante la resolución de problemas reales que reflejan las percepciones de los docentes.

## AGRADECIMIENTOS

Investigación y desarrollo comprendido en los PI Sistemas informáticos: modelos, métodos y herramientas (PI 19F014, Res. Nº 1015/19 CS.)

y Sistemas informáticos y gestión del conocimiento. Modelos, métodos y herramientas (PI 22F025, Res. Nº 0931/22 CS UNNE). Acreditado por la Secretaría General de Ciencia y Técnica, Universidad Nacional del Nordeste.

## REFERENCIAS BIBLIOGRÁFICAS

- Abran, A., Yurkov, A., Khalin, V. y Shilova, O. (2022). Quantitative analysis of informational significance of SWEBOK knowledge areas in IEEE/ACM curriculum guidelines. *In System Analysis in Engineering and Control* (pp. 561-573). Cham: Springer International Publishing. [https://www.researchgate.net/publication/359998231\\_Quantitative\\_Analysis\\_of\\_Informational\\_Significance\\_of\\_SWEBOK\\_Knowledge\\_Areas\\_in\\_IEEEACM\\_Curriculum\\_Guidelines](https://www.researchgate.net/publication/359998231_Quantitative_Analysis_of_Informational_Significance_of_SWEBOK_Knowledge_Areas_in_IEEEACM_Curriculum_Guidelines)
- Alenezi, M. y Akour, M. (2022). Methodical Software Testing Course in Higher Education. *International Journal of Engineering Pedagogy*, 12(1). [https://www.researchgate.net/publication/358504537\\_Methodical\\_Software\\_Testing\\_Course\\_in\\_Higher\\_Education](https://www.researchgate.net/publication/358504537_Methodical_Software_Testing_Course_in_Higher_Education)
- Arias, M., Tomaselli, G. P. y Tortosa, N. (2021). Computación afectiva y agilidad en entornos virtuales de educación: una revisión sistemática de la literatura. *VIII Simposio Argentino sobre Tecnología y Sociedad (STS 2021) - JAIIO 50* (Modalidad virtual). Argentina. <https://www.sciencedirect.com/science/article/abs/pii/S0360131519302027>
- Botella Nicolás, A. M. y Ramos Ramos, P. (2020). La relación con los demás y la motivación en un Aprendizaje Basado en Proyectos. *Estudios pedagógicos (Valdivia)*, 46(1), 145-160. <https://dx.doi.org/10.4067/S0718-07052020000100145>

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>



<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

- Bourque, P. y Fairley, R. (2014). SWEBOK Guide to the Software Engineering Body of Knowledge, Version 3.0. IEEE Computer Society.  
<https://www.computer.org/education/bodies-of-knowledge/software-engineering>
- Capuña y Uceda, O. E. y Antón Pérez, J. M. (2021). Influencia de SCRUM en los plazos de entrega y rendimiento en los proyectos de las asignaturas de Desarrollo de Software. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, no. 29, 36-42.  
[https://www.scielo.org.ar/scielo.php?script=sci\\_arttext&pid=S1850-99592021000200005](https://www.scielo.org.ar/scielo.php?script=sci_arttext&pid=S1850-99592021000200005)
- Hernández Sola, G. (2021). Nuestro análisis de la agilidad en 2021 a partir del 15th Annual State of Agile Report.  
<https://www.Scrum.org/resources/blog/nuestro-analisis-de-la-agilidad-en-2021-partir-del-15th-annual-state-agile-report>
- Kamthan, P. y Washizaki, H. (2022). Educational and Professional Implications of SWEBOK. *Proceedings of the 34th International Conference on Software Engineering and Knowledge Engineering. SEKE 2022*. KSIR Virtual Conference Center Pittsburgh, USA.  
<https://worldscientific.com/doi/10.1142/S0218194022500590>
- Romillo, A. y Polaino, C. (2019). Aplicación del Modelo de Gestión Pirámide del Desarrollo Universitario en la Universidad de Otavalo, Ecuador. *Formación Universitaria*, 12(1), 3-12.  
[https://www.scielo.cl/scielo.php?script=sci\\_arttext&pid=S0718506201900010003](https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718506201900010003)
- Ruiz de Mendarozqueta, A., Bustos, F. y Colla, P. (2022). Relationship between mature software engineering practices and agility practices. *Electronic Journal of SADIO*. 21(2). 79-109.  
[https://sedici.unlp.edu.ar/bitstream/handle/10915/142904/Documento\\_completo.pdf-PDFA.pdf?sequence=1](https://sedici.unlp.edu.ar/bitstream/handle/10915/142904/Documento_completo.pdf-PDFA.pdf?sequence=1)
- Soriano, K., Rosero, P., Guzmán, J. y Nieves, Z. (2024). Implementación en el aprendizaje basado en proyectos (ABP) en la educación universitaria: impacto en la motivación y el rendimiento de los estudiantes. *Revista Social Fronteriza*. 4. e45456.  
<https://www.revistasocialfronteriza.com/ojs/index.php/rev/article/view/475>
- Vila Grau, J. L. y Capuz-Rizo, S. (2021). La extensión de la agilidad al ámbito docente a través de Scrum & EduScrum. *In 25th International Congress on Project Management and Engineering At: Alcoi (Spain)*. 1, 2276-2290, Project: PhD research Project of Juan Luis Vila Grau.  
<http://dspace.aepro.com/xmlui/handle/123456789/3047>
- Villanueva Morales, C., Ortega Sánchez, G. y Díaz Sepúlveda, L. (2022). Aprendizaje Basado en Proyectos: metodología para fortalecer tres habilidades transversales. *Revista de estudios y experiencias en educación*, 21(45), 433-445.  
<https://doi.org/10.21703/0718-5162.v21.n45.2022.022>

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>

**Conflicto de intereses:**

Los autores declaran no tener conflictos de intereses.

**Contribución de los autores:**

Los autores participaron en el diseño y redacción del artículo, en la búsqueda y análisis de la información contenida en la bibliografía consultada.

**Citar como**

Mariño, S. I., Alfonzo, P. L. (2024). Modelos, simuladores e Ingeniería del Software. *Mendive. Revista de Educación*, 22(4), e3599.

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>



Esta obra está bajo una [licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/)

<https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/3599>