

Scrum como Herramienta Metodológica para el Aprendizaje de la Programación

Nicolás Tymkiw, Juan Manuel Bournissen y Marisa Cecilia Tumino
Universidad Adventista del Plata, Libertador San Martín, Entre Ríos, Argentina
{Nicolás.tymkiw, juan.bournissen, marisa.tumino}@uap.edu.ar

Resumen. El objetivo de este estudio fue identificar el impacto de la utilización de la metodología de desarrollo de software Scrum, como una técnica que permite reforzar las estrategias de enseñanza y aprendizaje utilizadas en la enseñanza de nivel superior en la asignatura de Programación II, en el aprendizaje de los estudiantes de una universidad privada.

El proyecto ha proporcionado datos suficientes que permiten estimar que la aplicación de la metodología Scrum tiene un impacto positivo como refuerzo en el proceso de aprendizaje de los estudiantes, ofreciendo una alternativa innovadora a las estrategias tradicionales utilizadas por las cátedras donde se realizó el experimento.

Palabras clave: Scrum en el aula, Aprendizaje de programación, Metodología, Impacto

1. Introducción

En la actualidad, dentro de las carreras informáticas, la programación es una asignatura fundamental, debido a que es una de las materias base que determina el progreso del alumno en la carrera y luego, una vez recibido, determina su desempeño profesional.

Investigadores como Costelloe [1] demuestran que los estudiantes no logran conceptualizar todos los temas de programación enseñados, siendo este uno de los puntos principales por el cual dejan la carrera. Otro aspecto que enfatizan los autores es que si logran pasar la materia más adelante, el rendimiento en asignaturas correlativas es muy bajo.

Esta metodología requiere de un proceso evaluativo, en el que se reveen las prácticas docentes, viendo los obstáculos y dificultades que se presentan en los estudiantes.

Barberis, y Del Moral Sachetti [2], señalaron que desde que se implementó esta metodología en el año 2014, la tasa de deserción bajó un 23% con respecto a los años anteriores, mientras que el rendimiento académico aumentó un 27%.

Perazo [3] afirma que el porcentaje de la deserción en las carreras informáticas se aproxima al 80%, teniendo en cuenta los datos suministrados por la Secretaría de Políticas Universitarias del Ministerio de Educación. Esta problemática puede deberse a que, en su mayoría, es una decisión que toman adolescentes entre 17 y 20 años que desconocen las incumbencias de la profesión elegida.

1.2. Beneficios de Scrum

Albaladejo [4] menciona que es necesario saber que SCRUM cuenta con beneficios, fundamentos y requisitos, entre los que cabe mencionar los que se detallan a continuación.

- Entrega de resultados que debe ser dentro de un plazo corto, mensual o quincenal, en el que los requisitos que son prioridad ya estén completados.
- Facilita la gestión de las expectativas del cliente mediante dos requerimientos, en el primero el cliente presenta una lista de requisitos, indicando su prioridad; en el segundo, al finalizar la iteración, el equipo le muestra al cliente los resultados de los requisitos que han completado.
- Utilización de las funciones más importantes antes de que el proyecto termine. El cliente puede empezar a recuperar su inversión con anterioridad, lo que se consigue mediante la priorización de requisitos por coste y valor. El Retorno de Inversión (ROI) es un beneficio que el cliente maximiza en el proyecto, si ve que el beneficio es menor al coste del desarrollo el cliente puede finalizar el proyecto.
- Proporciona flexibilidad y adaptación, donde el cliente va dirigiendo el proyecto en función de sus nuevas prioridades, lo que se consigue replanificando los requisitos y prioridades en el inicio de cada iteración.
- Desde la primera iteración el equipo debe definir los riesgos para poder mitigarlos, si hay que equivocarse, es mejor hacerlo lo antes posible. Para lidiar con este beneficio es necesario un desarrollo iterativo e incremental, de esta manera no se deja para el final ninguna actividad riesgosa vinculada con la entrega de requisitos.
- Productividad y calidad es otro de los beneficios que proporciona SCRUM, el equipo mejora y simplifica su manera de trabajar, para esto es necesario que en cada iteración el equipo realice una retrospectiva para analizar su método de trabajo, como así también se necesita una comunicación continua del equipo.
- Alimentación entre cliente y equipo, para lo que se necesita que ambos sean un equipo, que trabajen juntos en los requisitos, detalles y análisis de los resultados obtenidos.
- Motivación del equipo, para lo que es importante dejar que las personas puedan usar su creatividad para resolver problemas y decidir cómo organizar su trabajo, por lo que es necesario un equipo autogestionado que se comprometa a completar todos los requisitos determinados en una iteración.

1.3. Componentes de Scrum

La metodología Scrum se encuentra dividida en fases y roles, a las fases las podemos identificar como reuniones, también conocidas como Sprint.

Reuniones. Gallego [5] divide las reuniones en tres fases: (a) en la primera fase se desarrolla una planificación del Backlog, dentro del cual se definen las prioridades de los requisitos y la planificación del primer sprint 0, con los objetivos y el trabajo que se necesita completar en esa iteración. Durante esta fase se obtiene la lista de tareas a realizar; (b) para la segunda fase se genera un seguimiento del sprint. Es una fase de reuniones diarias en las que se evalúa el lugar del proyecto en el que se encuentra el equipo, los avances desde la última reunión, los trabajos que se realizarán hasta la siguiente reunión y se analiza la forma de solucionar los inconvenientes que surgieron durante el desarrollo y (c) en la tercera fase, cuando concluye el Sprint, se aplica una revisión del incremento que se ha generado, se presentan los resultados finales y una versión de ayuda para mejorar la realimentación del cliente..

Roles. Los roles se dividen en dos grupos: (a) los que están comprometidos con el proyecto y proceso de Scrum y (b) los que no son parte del proceso pero que se necesitan para la realimentación de la salida de los procesos y el planeamiento de los sprint. En el primer grupo se pueden identificar a los integrantes en tres áreas:

Product Owner: es el encargado de tomar las decisiones del proyecto y es el conocedor del negocio del cliente y su visión sobre el producto. Es responsable de recolectar los requisitos del cliente y ordenarlos por prioridad.

Scrum Master: comprueba que la metodología funciona, para lo que debe sobrepasar los inconvenientes que obstaculicen la fluidez del proceso e interactuar con el cliente y los gestores.

Equipo de Desarrollo: conformado por un grupo pequeño de personas y tienen la autoridad para organizar y tomar decisiones para conseguir su objetivo.

Dentro del segundo grupo se pueden encontrar los siguientes roles:

Usuarios: son los destinatarios finales del producto.

Stakeholders: son los que participan de las revisiones del sprint.

Managers: encargados de la toma de decisiones finales, participan en la selección de los objetivos y los requisitos del proyecto..

1.4. Scrum en el aula

Kuz, Falco y Giandini [6] señalan que, dentro del ámbito educativo, los estudiantes necesitan desarrollar capacidades y aptitudes que posteriormente le servirán en su carrera profesional. Las competencias generales requieren de métodos de aprendizaje activos que posibiliten el desarrollo de la capacidad de organización, planificación, liderazgo, evaluación, autoevaluación y trabajo en equipo, entre otros.

Teniendo en cuenta lo mencionado anteriormente, Scrum favorece la creación de un ambiente propicio para que los estudiantes puedan experimentar un aprendizaje creativo, enriquecedor y confiable.

Miller [7] explica que un “aula ágil” precisa de la integración de cinco elementos fundamentales para alcanzar objetivos específicos: (a) clase visible, (b) ritmo de

aprendizaje, (c) colaboración, (d) capacitación y (e) el camino para evolucionar un aula hacia la autoorganización.

1.5. Evaluación de Impacto

Tejada Fernández [8] explica que la evaluación del impacto intenta responder la siguiente pregunta. ¿Cuáles son los efectos que la formación produce en un determinado ámbito? Se intenta averiguar en qué medida los aprendizajes adquiridos en las acciones formativas son útiles para mejorar el desempeño.

El proceso de evaluación de acciones formativas cuenta con tres finalidades esenciales: (a) diagnóstica, que permite conocer el desarrollo del proceso de enseñanza y aprendizaje, (b) formativa, que permite valorar una acción educativa durante su desarrollo y (c) sumativa o de productos, que permite conocer si los objetivos se consiguieron o no, los cambios que se produjeron y la toma de decisiones con respecto a las actividades planteadas.

1.6. Scrum en programación

Varios autores muestran que la utilización de metodologías ágiles, como Scrum, pueden ser altamente eficientes al desarrollar un software. A continuación, se presentan investigaciones que avalan lo expuesto.

Chávez Andrade [9] indagó sobre la estandarización de los procesos de desarrollo utilizando buenas prácticas de programación y Scrum en departamentos de TI. Tuvo como objetivo incrementar el grado de funcionalidad de un producto de software. Afirma que la carente aplicación metodológica conlleva a obtener un producto de baja calidad. Explica que la aplicación de Scrum logra la integración del usuario al proceso de desarrollo, así como genera un ambiente de compromiso en el equipo de desarrollo con el fin de obtener un producto de software de calidad, que cumpla con las aspiraciones del usuario. Se identificó la mejora de funcionalidad como atributo de la calidad por parte de los implicados en el proyecto.

Hervás Lucas [10] sostiene que, para alcanzar un aprendizaje eficiente en un lenguaje de programación, se emplean técnicas de integración continua. La investigación del autor fue realizada con alumnos del tercer curso de ingeniería informática. El aprendizaje con este método se basa en el desarrollo guiado y supervisado de prototipos de complejidad incremental. La conclusión a la que llega el autor es que el método prototipado ágil fomenta el aprendizaje y el desarrollo de competencias clave como el trabajo en equipo, la autodidáctica y el análisis crítico.

2. Metodología

El presente estudio es de tipo explicativo y cuasiexperimental. Explicativo porque busca determinar el nivel de impacto en dos grupos que utilizaron una metodología de estudio diferente, y cuasiexperimental porque para la experiencia se tomaron los grupos tal como estaban conformados. El diseño metodológico estuvo basado en un estudio realizado en la Universidad Nacional de Salta por Barberis y otros, [2].

La variable independiente en esta investigación fue la metodología utilizada y las variables dependientes: (a) el nivel de impacto en el aprendizaje de los estudiantes y (b) las calificaciones promedio cuatrimestrales.

2.1. Implementación

Durante los últimos años se ha identificado que en la asignatura de Programación II existía un alto porcentaje de alumnos que perdían la materia y otro porcentaje, menor pero significativo, de alumnos que decidían abandonarla.

En base a esta problemática se decidió desarrollar un plan de trabajo con el fin de que los alumnos puedan obtener todos los conocimientos que estipula la asignatura y que la implementación de estos se vea reflejado en las calificaciones.

Este plan de trabajo consistió en aplicar la metodología Scrum para el aprendizaje de la programación. Se dividió al curso en grupos de tres, y se les explico a los estudiantes la modalidad de trabajo del año. Al dividir al curso en grupos lo que se buscó fue que a través del trabajo en grupo se pueda potenciar las habilidades de cada uno de los estudiantes.

Durante el cursado de la asignatura, a medida que se incrementaban los conceptos, se les asignaron trabajos prácticos. Los estudiantes debían cumplimentarlos en un plazo de tres semanas, atendiendo a una serie de requisitos:

- a. La composición del grupo durante un trabajo practico debería contar con un líder de Scrum, mientras que los demás estudiantes asumirían el rol de equipo de desarrollo.
- b. Cada uno de los estudiantes tendría que pasar por el rol de líder de Scrum.
- c. El equipo debería realizar al menos un sprint (reunión de equipo) por semana, donde mediante un brainstorming podrían elegir como desarrollar los requisitos propuestos o solucionar los problemas que fueron apareciendo durante el transcurso del desarrollo.
- d. Cada semana el equipo se reuniría con el profesor de la catedra, quien asumiría el rol de cliente e iría revisando si el sistema cumplía con los requisitos que se habían planteado e incrementando de acuerdo con los nuevos conceptos vistos en clase.
- e. En la tercera semana los equipos deberían exponer a toda la clase el práctico que se les había asignado. En esta exposición debían describir la modalidad que adoptaron para desarrollarlo, los problemas con que se encontraron y las soluciones encontraron para avanzar. Por su parte el líder Scrum debía relatar la forma en que se realizaron los sprints y como fue el trabajo del grupo con el cliente. Por último,

cada uno de los estudiantes del grupo debía mostrar la parte del proyecto que había desarrollado y explicar porque eligió desarrollarlo de esa manera. Esto último serviría para observar si el alumno realmente había podido comprender los conceptos aprendidos en clase.

Además de los trabajos prácticos, las calificaciones se tomaron en base a dos evaluaciones. La primera constó de un examen individual en el que los estudiantes desarrollaron un sistema con los conceptos aprendidos hasta el momento. La segunda evaluación tuvo una temática distinta a la primera en el que el profesor le asignó a cada grupo una problemática distinta y les concedió tres horas para que en equipo elaboraran un sistema que solucionara dicha problemática.

La cursada finalizó con un trabajo final, con un grado de dificultad mayor al resto de los prácticos realizados, en el que se integraban todos los conceptos vistos durante el cuatrimestre. Para calificarlo se les pidió a los equipos que, además de cumplir con cada uno de los requerimientos, le agregaran funcionalidades extras y que se le dedique tiempo al diseño del software.

2.2. Sujetos

La muestra estuvo compuesta por sujetos que cursan su carrera de sistemas de información en una universidad de la provincia de Entre Ríos. Se evaluaron dos grupos de estudiantes:

El primer grupo estuvo conformado por estudiantes de ambos géneros, entre 18 y 25 años, que implementaron la metodología Scrum, en la asignatura de programación II de la carrera de Ingeniería en Sistemas de la Información, durante el segundo cuatrimestre del año 2018.

El segundo grupo se conformó por estudiantes, de género masculino, entre 22 y 26 años, que habían cursado en años anteriores al 2018 la asignatura de programación II bajo la implementación de metodologías distintas a la metodología del primer grupo.

El profesor titular y el profesor adjunto de la cátedra de programación II fue el mismo en ambos grupos. Los docentes, además de enseñar los conceptos estipulados, asumieron el rol de clientes para cada uno de los trabajos prácticos asignados.

2.3. Instrumentos

Se midió el impacto mediante el test de Nivel de Impacto de Tumino y Bournissen [11]. El test presenta una estructura compuesta por 8 variables vinculadas al aprendizaje y 10 variables relacionadas con estrategias de aprendizaje. La escala utilizada en las dos dimensiones es de intervalo desde 1 = muy en desacuerdo, 2 = en desacuerdo, 3 = neutro, 4 = de acuerdo y 5 = muy de acuerdo.

Esta encuesta fue aplicada a los estudiantes que cursaron la asignatura de programación II con la metodología Scrum como estrategia de aprendizaje y a estudiantes que cursaron con metodologías previas.

2.4. Procedimientos para recolección de datos

En primer lugar, se realizó una reunión con la dirección de la carrera y el profesor titular de la cátedra para informar sobre el propósito y las características de la investigación. En dicha reunión se ofreció un modelo a seguir que consta de un trabajo experimental en el que se evaluaría el aprendizaje de los estudiantes, con la finalidad de disminuir la tasa de deserción o pérdida de la asignatura respecto de años anteriores.

El docente sugirió el modelo de armado de equipos, a partir de una evaluación inicial niveladora, a fin de asegurar el beneficio de todos los estudiantes.

Durante los meses de cursado se asistió constantemente a las clases prácticas para poder compartir observaciones acerca del trabajo realizado por los estudiantes y constatarlo en una bitácora de anotaciones.

Finalizada la cursada se les administró a los estudiantes los test para recolectar los datos de nivel de impacto en su aprendizaje. También se le aplicó el test a estudiantes que años anteriores habían cursado la asignatura con otras metodologías.

Por último, se realizó una reunión con los profesores para conocer sus percepciones en cuanto a la metodología aplicada y se obtuvieron las calificaciones de quienes respondieron el test de nivel de impacto. Asimismo, se obtuvieron los datos de cursado de las cohortes de los últimos diez años a fin de analizar comparativamente los resultados.

2.5. Procedimientos para el análisis estadístico de los datos

Una vez recolectados los datos, se utilizó el programa IBM SPSS Statistics versión 23.0 para el análisis descriptivo y comparativo de los datos.

Se trabajó con dos pruebas no paramétricas mediante las que se compararon el nivel de impacto en el aprendizaje y la calificación promedio entre los estudiantes que utilizaron la metodología SCRUM, como estrategia de enseñanza y aprendizaje, y los que aprendieron mediante otras estrategias anteriormente.

3. Resultados

A continuación, se muestran los resultados obtenidos a partir de las variables presentadas en la investigación.

El objetivo principal fue identificar si los estudiantes que trabajaron con la metodología Scrum, podrían mostrar un mejor nivel de impacto en el aprendizaje percibido de la

asignatura, lo que se reflejaría tanto en las calificaciones promedio, como en el impacto de la implementación de la metodología en su aprendizaje.

3.1. Datos de cursado

Mediante el análisis realizado con los datos de cursado de la asignatura obtenidos de las últimas diez cohortes, se pudo observar que en el año en que se realizó la experiencia, el 66,67% de los estudiantes promocionó la asignatura mientras que el 33,3% restante la regularizó. De esta manera ningún estudiante perdió o abandonó la asignatura.

De los últimos diez años de cursado de la asignatura, los nueve años anteriores se utilizó una misma metodología y en el último año se implementó la metodología Scrum. Se observa que tan solo en este último año ningún alumno perdió o abandonó la cursada, tal como se muestra en la Tabla 1.

Tabla 1: Datos de las calificaciones de los últimos diez años.

Año	Promoción Directa	Promoción Indirecta	Pierde la Materia	Abandona
2009	30,0%	30,0%	30,0%	10,0%
2010	11,8%	41,2%	35,3%	11,8%
2011	33,3%	40,0%	26,7%	0%
2012	20,8%	37,5%	33,3%	8,3%
2013	10,0%	50,0%	40,0%	0%
2014	29,4%	23,5%	41,2%	5,9%
2015	28,6%	28,6%	42,9%	0,0%
2016	9%	18,2%	54,6%	18,2%
2017	28,5%	52,4%	19,1%	0%
2018	66,7%	33,3%	0%	0%

3.2. Desempeño de los equipos

Los resultados acerca del desempeño de los equipos se pudieron observar mediante la bitácora de observaciones. En esta bitácora se plasmó el trabajo de cada uno de los equipos durante la cursada de la asignatura de programación II, utilizando la metodología SCRUM como estrategia de enseñanza y aprendizaje. Se pudo observar que los equipos mantuvieron un alto desempeño, buena organización, y concluyeron con los proyectos en el tiempo estipulado, en la medida en que cumplieron con las pautas de trabajo de la metodología. También se observó que desarrollar proyectos, dentro de un ámbito de equipo como exige la metodología Scrum, generó un ambiente de compromiso y potenció la competitividad de los estudiantes en cuanto a los requisitos que debían cumplimentar en cada uno de los proyectos. Otro resultado positivo que arrojó el trabajo con la metodología Scrum fue que les permitió a los estudiantes de segundo año de la carrera de Ingeniería en Sistemas, defender los proyectos oralmente y adquirir un lenguaje técnico apropiado.

3.3. Pruebas de hipótesis

A continuación, se presentan los resultados concernientes a las hipótesis planteadas durante la investigación.

3.3.1. Primera hipótesis

H1: Existe diferencia estadística significativa de rangos promedios de rendimiento académico, en términos de calificación promedio, obtenido en programación II entre los estudiantes que utilizan la estrategia de enseñanza Scrum y los que cursaron con las metodologías utilizadas previamente.

Esta hipótesis se probó mediante las pruebas no paramétricas para dos muestras independientes tomando a la calificación final como variable de prueba y a la metodología como variable de agrupación. En base a esos datos se observó que existe una diferencia estadísticamente significativa de rango promedio de calificación promedio entre los dos grupos ($p < 0,05$). Se encontró un mayor rango promedio de calificación promedio en los estudiantes que trabajaron con la metodología Scrum (RP = 9.93), m. Mientras que los estudiantes que trabajaron con otras metodologías obtuvieron un menor rango promedio (RP = 5.07).

3.3.2. Segunda hipótesis

H2: Existe diferencia estadística significativa en el nivel de impacto, en el aprendizaje percibido, obtenido en programación II entre los estudiantes que utilizan la metodología SCRUM como estrategia de enseñanza y aprendizaje y los que cursaron con las estrategias utilizadas previamente.

En esta hipótesis se analizaron los datos obtenidos de las muestras mediante las pruebas no paramétricas para dos muestras independientes y se encontró un rango promedio en la categoría aprendizaje más elevado (ver Tabla 2) que demostró una diferencia estadísticamente significativa del rango promedio. Se encontró un mejor nivel de impacto en el aprendizaje percibido ($p < 0,05$) y un rango promedio más elevado (RP = 10.07) para los estudiantes que utilizaron la metodología Scrum, mientras que los estudiantes que usaron otras metodologías obtuvieron un rango promedio de 4,93.

En cuanto a la categoría estrategia de aprendizaje percibido del test del Nivel del Impacto, no se obtuvo una diferencia estadísticamente significativa, pero si se obtuvo una diferencia en el rango promedio entre las dos metodologías. Con la metodología Scrum el rango promedio fue mayor (RP = 9.5) que con las otras metodologías (RP = 5.5) como se observa en la Tabla 2.

Tabla 2: Diferencia entre rango promedio de aprendizaje y estrategia-aprendizaje.

Categoría	Metodología	Rango promedio	Suma de rangos
Aprendizaje	Scrum	10,07	70,5
	Otras	4,93	34,5
Estrategia-Aprendizaje	Scrum	9,5	66,5
	Otras	5,5	38,5

4. Discusión y Conclusión

En el presente apartado se exponen la discusión y conclusión de la investigación, como así también se presentan recomendaciones para futuras investigaciones.

4.1. Discusión

Mediante el análisis de los datos obtenidos, se pudo observar que existe una diferencia estadísticamente significativa de rango promedio en el aprendizaje percibido entre los dos grupos estudiados. También existió una diferencia significativa de rango promedio de la calificación final de los alumnos que utilizaron la metodología Scrum, respecto de quienes trabajaron con otras metodologías.

Otro aspecto que se observó fue que en el transcurso de los años 2009 al 2017, existía un predominio en la promoción indirecta, e incluso en la pérdida de la materia, mientras que, en el año 2018, con la implementación de la metodología Scrum, se denotó el predominio de la promoción directa de los estudiantes.

Los estudiantes que tuvieron la posibilidad de trabajar con Scrum mostraron un mayor impacto mediante sus respuestas. Los docentes, que se habían mostrado abiertos ante la propuesta de implementar la nueva metodología, expresaron entusiasmo con los resultados obtenidos y decidieron mantenerla durante los próximos años.

Por otra parte, en las encuestas realizadas a los estudiantes que cursaron con metodologías anteriores se observó, a través de sus respuestas, que no se encontraban muy convencidos con la metodología utilizada. Se estimó que este resultado pudo darse por distintos factores, algunos de ellos pueden haber sido por la falta de comprensión sobre los temas dados en clase, la falta de compromiso con la asignatura, o la falta de comprensión de la metodología aplicada.

Es por esto que el objetivo principal fue medir y mostrar el nivel de impacto que causó la utilización de la metodología Scrum en el aprendizaje percibido y en la estrategia de enseñanza dentro del ámbito académico.

4.2. Conclusión

La utilización de la metodología Scrum en el ámbito académico, y específicamente dentro de las asignaturas de programación, donde los estudiantes pueden aplicarla e integrarla a los proyectos que les sean asignados, resulta una práctica muy satisfactoria ya que causa un impacto positivo en el aprendizaje de los estudiantes.

En la investigación realizada el impacto causado en los alumnos fue positivo tanto en las calificaciones como en el aprendizaje percibido al tener que enfrentarse a las nuevas propuestas metodológicas. Los estudiantes lograron captar los contenidos establecidos por la metodología, tales como el desarrollo de las actividades de forma grupal, los tiempos de entregas, la interpretación de roles y la exposición sobre el trabajo realizado. También se observó mediante la bitácora de observaciones que los estudiantes pudieron trabajar de la mejor manera al momento de establecer las reuniones de grupo, determinar los sprints que iban a realizar para obtener el resultado exigido y poder sobreponerse a las dificultades que se presentaban a medida que desarrollaban el producto.

La implementación de la metodología Scrum facilitó a los docentes el mantenimiento de la dinámica de la clase, consiguiendo que todos los estudiantes pudiesen realizar un seguimiento constante de los contenidos presentados en la asignatura.

Por otra parte, el impacto causado en los estudiantes fue mayor con la metodología Scrum obteniendo un rango promedio de 10,07 y 9,5 en cuanto al aprendizaje percibido y la estrategia de aprendizaje respectivamente.

Con la implementación de la metodología Scrum, se observó mayor compromiso de parte de los estudiantes para cumplir con las directrices de la asignatura, al generar un ambiente de trabajo ideal dentro del aula donde los estudiantes pueden aprovechar los conocimientos que les proporcionan los docentes y sus pares.

5. Referencias

1. Costelloe E. (2004). "Teaching Programming The State of the Art. CRITE. Technical Report". Department of Computing, Institute of Technology Tallaght, Dublin, Ireland.
2. Barberis, Á. R., & Moral Sachetti, L. E. D. (2016, August). Scrum como herramienta metodológica en el entrenamiento cooperativo de la programación: de la teoría a la práctica. In XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016), recuperado de: http://sedici.unlp.edu.ar/bitstream/handle/10915/54603/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y:
3. Perazo C. (25/10/2013) Tecnología. Reporte del Ministerio de Educación. Diario La Nación. Recuperado de: <http://www.lanacion.com.ar/1632045-el-80-de-los-estudiantes-de-carreras-informatica>
4. Albaladejo X. (2008) Fundamentos de scrum. Recuperado de <https://proyectosagiles.org/fundamentos-de-scrum/>
5. Gallego, M. T. (2012). Metodologia scrum. Universitat Oberta de Catalunya.

6. Kuz, A., Falco, M., & Giandini, R. S. (2018). Comprendiendo la aplicabilidad de Scrum en el aula: herramientas y ejemplos. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (21), e07-e07.
7. Miller J. (2016, 1 de noviembre), 5 Elements of Agile Classrooms. Recuperado de <http://blog.agileclassrooms.com/2016/11/5-elements-of-agile-classrooms.html>
8. Tejada Fernández, J., & Ferrández Lafuente, E. (2007). La evaluación del impacto de la formación como estrategia de mejora en las organizaciones. *Revista electrónica de investigación educativa*, 9(2), 1-15.
9. Chávez Andrade, J. V. (2019). Estandarización de los procesos de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil en departamentos de TI (Master's thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Maestría en Gerencia de Sistemas de Información).
10. Hervás Lucas, R. (2012). Método para el aprendizaje de entornos y lenguajes de programación basado en prototipado ágil. *Jornadas de Enseñanza de la Informática (18es: 2012: Ciudad Real)*.
11. Tumino, M. C. y Bournissen, J. M., (2020). Integration of Information and Communication Technologies (ICT) in the classroom and its impact on students: construction and validation of measurement scales. *Journal of Educational Research and Innovation (IJERI)*, 13, en Prensa.