

Enseñando Scrum desde Diversas Perspectivas de la Ingeniería Informática

Verónica Romero^a, Miriam Gil^b y Jose Ignacio Panach^c

^aEscola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València, verogo@uv.es 

^bEscola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València, miriam.gil@uv.es 

^cEscola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València, joigpana@uv.es 

How to cite: Verónica Romero, Miriam Gil y Jose Ignacio Panach. 2023. Enseñando Scrum desde Diversas Perspectivas de la Ingeniería Informática. En libro de actas: *IX Congreso de Innovación Educativa y Docencia en Red*. Valencia, 13 - 14 de julio de 2023. Doi: <https://doi.org/10.4995/INRED2023.2023.16626>

Abstract

Agile methodologies are being increasingly used in software development projects. Among the most widely used agile methodologies, the Scrum methodology stands out. Given the importance of these methodologies, there are several subjects related to software development and project management in which they are taught and used. This article presents three different subjects related to the Computer Engineering Degree (GII) and the Master's Degree in Web Technologies, Cloud Computing and Mobile Applications (TWCAM) where the Scrum methodology has been incorporated.

Keywords: *Agile methodologies, Scrum methodology, software engineering, project management, software production*

Resumen

Las metodologías de desarrollo ágiles están siendo cada vez más utilizadas en proyectos de desarrollo de software. Entre las diferentes metodologías ágiles más utilizadas destaca la metodología Scrum. Dada la importancia de dichas metodologías, son varias las asignaturas relacionadas con el desarrollo de software y la gestión de proyectos en las que se enseñan y se utilizan. En este artículo se presentan tres asignaturas distintas relacionadas con el Grado de Ingeniería Informática (GII) y el Máster Universitario en Tecnologías Web, Computación en la Nube y Aplicaciones Móviles (TWCAM) donde se ha incorporado la metodología Scrum.

Palabras clave: *metodologías ágiles, metodología Scrum, ingeniería del software, gestión de proyectos, producción de software*

1. Introducción

Las metodologías de desarrollo ágiles son métodos de desarrollo software basados en un proceso iterativo e incremental. Bajo esta filosofía, los requerimientos y soluciones evolucionan por medio de la colaboración de equipos multifuncionales autoorganizados (Cockburn, 2006). Son varias las ventajas que se engloban dentro de estas metodologías: i) dividir el trabajo reduce riesgos; ii) garantizan la satisfacción del cliente porque éste es parte del equipo; iii) tiene la capacidad de respuesta a posibles cambios; iv) prevalece la

simplicidad; v) en grupos pequeños de desarrollo la comunicación es mejor; vi) a la hora de la entrega final toda ha sido probado; vii) ahorra tiempo quitando actividades innecesarias.

Dentro de las diversas metodologías ágiles que se utilizan actualmente, una de las utilizadas en la industria es Scrum (Schwaber & Sutherland, 2011). Scrum es una metodología basada en el ciclo de Deming de un desarrollo mantenido e incremental, haciendo hincapié en el trabajo en equipo por encima del individualismo. Se basa en 4 roles: i) Scrum máster: es el experto de scrum y el que vela por aplicar correctamente la metodología de desarrollo; ii) product owner: es alguien del equipo de desarrollo que conoce bien el negocio donde se va a poner en marcha el sistema; iii) desarrolladores: son los programadores que escriben el código del sistema; iv) stakeholder: es cualquier persona interesada en el sistema en desarrollo. Todos los requisitos a abarcar se definen en el product backlog, y se abordan en ciclos de desarrollo cortos, denominados sprints. En cada sprint, se seleccionan los requisitos a implementar (sprint backlog) y se hace análisis, diseño e implementación.

La contribución de este artículo es exponer cómo hemos incorporado Scrum a tres asignaturas distintas relacionadas con el Grado de Ingeniería Informática (GII) y el Máster Universitario en Tecnologías Web, Computación en la Nube y Aplicaciones Móviles (TWCAM) de la Universitat de València. Aunque en todas las asignaturas se aboga por Scrum, el contexto de uso de esta metodología es distinto. Las asignaturas implicadas son: i) Ingeniería del Software II (3ro de GII); ii) Gestión de Proyectos (3ro de GII); iii) Métodos de Producción de Software (TWCAM). En este artículo vamos a abordar cómo se ha utilizado Scrum como método de ejercicios prácticos y se van a evaluar los resultados en base a los proyectos entregados de forma satisfactoria.

2. Objetivos

Las 3 asignaturas se enmarcan en la materia de la Ingeniería del Software, y por tanto las competencias a alcanzar son transversales en todas ellas. Los objetivos que se pretenden alcanzar con el uso de Scrum en las 3 asignaturas son:

- **(O1)** Que los/las estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.
- **(O2)** Que los/las estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.
- **(O3)** Capacidad para la elaboración, planificación, dirección, coordinación, gestión técnica y económica y la implantación de proyectos software.
- **(O4)** Capacidad para aplicar metodologías de Ingeniería del Software en el desarrollo y gestión de un proyecto.

3. Desarrollo de la Innovación

3.1 Scrum en Ingeniería del Software II

En la asignatura de Ingeniería del Software II se introduce la metodología Scrum para el desarrollo de proyectos software. El objetivo de aprendizaje es que el alumnado adopte y aplique las técnicas de desarrollo ágil trabajando en equipos Scrum sobre un proyecto software. El proyecto se desarrolla a lo largo de las 8 sesiones de laboratorio, en el que el alumnado, en base a un enunciado, define las historias de usuario (requisitos del sistema), elabora la pila de producto, y planifica tres sprints para el desarrollo del proyecto. Para que el alumnado desarrolle las competencias necesarias del desarrollo ágil, en las clases de

teoría se aplica un aprendizaje basado en juegos. El uso de dinámicas de juegos en la docencia sobre agilidad en equipos de desarrollo de software es una práctica extendida (Guarino de Vasconcelos et al., 2018). Los juegos se seleccionan en función de los objetivos de aprendizaje. Los juegos propuestos en el curso 2021-2022 y que pretenden alcanzar los objetivos planteados en la Sección 2 han sido:

- Cartas de roles y valores: Este juego tiene como finalidad que los estudiantes discutan sobre los roles y valores de Scrum, instruyéndoles sobre las responsabilidades de cada rol y los comportamientos esperados. El estudiantado se divide en equipos (4 a 6 personas), a cada equipo se le reparte una baraja de cartas y estos deben encontrar la ubicación correcta de las cartas en base a una plantilla. El equipo que lo consigue primero gana. Las instrucciones detalladas del juego se pueden encontrar en (Aguila et al., 2023b).
- Juego de las monedas: La finalidad de este juego es mostrar la importancia del trabajo por etapas y las diferencias entre el desarrollo por etapas y el incremental. Los estudiantes se dividen en equipos (6 a 8 personas) y a cada equipo se le reparten 20 monedas. El trabajo del equipo consiste en procesar las monedas dándoles la vuelta con la mano no dominante de cada participante. Cada participante representa una etapa en el desarrollo (análisis, diseño, etc.). Un miembro del equipo se encarga de medir los tiempos que tardan en procesar todas las monedas. El juego se divide en varias rondas y en cada una de ellas se realiza un tipo de procesamiento de monedas (predictivo, ágil, etc.) para que los alumnos comprueben los beneficios del trabajo ágil, en cuanto a velocidad y aprovechamiento de recursos. Las instrucciones detalladas del juego se pueden encontrar en (Aguila et al., 2023a).
- Planning poker: El objetivo de este juego es aprender a realizar estimaciones relativas mediante la técnica de “planning poker”. En este juego los estudiantes se dividen en equipos (5 personas) y realizan 7 rondas de estimaciones. Para realizar la estimación se descargan una app móvil llamada “Scrum Poker” o similar. Las estimaciones a realizar son sobre el peso de animales. En la primera ronda se les muestra el animal más liviano (1 punto de historia), de forma que en las siguientes rondas deben estimar el peso relativo del resto de animales en comparación con el animal más liviano. Al final de la actividad, se revisan las estimaciones de los equipos para generar discusión entre todos los alumnos. Las instrucciones detalladas del juego se pueden encontrar en (Varela, 2023).

3.2 Scrum en Gestión de Proyectos

La metodología Scrum es una de las metodologías ágiles más utilizadas en el desarrollo de proyectos software. En la asignatura de Gestión de Proyectos se introduce dicha metodología y se presentan diferentes herramientas para la gestión de proyectos ágiles. El objetivo es que el alumnado adquiera las habilidades y los conocimientos necesarios para gestionar este tipo de proyectos.

La metodología docente utilizada para alcanzar los objetivos descritos en la sección 2 se basa, por un lado, en clases magistrales, donde el profesorado introduce nuevas técnicas y conceptos, y en clases de ejercicios prácticos, donde el estudiantado tiene que aplicar los conceptos y las técnicas estudiadas.

Por otro lado, también se utiliza la metodología basada en proyectos (Morales & Landa, 2004). A lo largo del curso, el alumnado desarrolla un proyecto en el que debe gestionar y planificar las diferentes iteraciones de un proyecto Scrum. Los estudiantes se dividen en grupos de 4 o 5. Uno de los miembros del grupo adquiere el rol de Scrum Master, mientras que el resto del equipo adquiere el rol de desarrolladores. Por su

parte, el profesor adquiere el rol de product owner. En cada iteración, dada una lista de historias de usuario, el equipo planifica el proyecto generando tanto el product backlog como el sprint backlog. Posteriormente, dada una situación que impide que la iteración se lleve a cabo tal y como se ha planificado, el equipo debe replanificar el sprint de la mejor forma posible. A continuación, se detallan los diferentes pasos del proyecto:

1. El profesor, en su rol de product owner, proporciona al equipo una lista de historias de usuario etiquetadas con el valor que cada una de ellas proporciona al cliente.
2. El equipo genera el product backlog y planifica que historias de usuario se llevarán a cabo en las diferentes iteraciones del proyecto. Para ello, en primer lugar, estima el esfuerzo en puntos de historia (PH) de cada una de las historias de usuario. Una vez calculados los PH y considerando el valor de cada historia se calcula el Retorno de la Inversión (ROI). Posteriormente, las historias se ordenan por prioridad según el valor del ROI obtenido. Finalmente, considerando la capacidad en puntos de historia que es capaz de llevar a cabo el equipo en un sprint, se genera el plan del proyecto, indicando cuántas iteraciones serán necesarias y que historias de usuario se realizarán en cada caso para proporcionarle el mayor valor al cliente.
3. El equipo genera el sprint backlog considerando las historias de usuario que se han planificado en la iteración actual. Para ello, cada una de las historias de usuario se divide en tareas más pequeñas y concretas, de forma que sea más fácil llevar a cabo la estimación de su esfuerzo.
4. El profesorado indica a cada equipo una situación que le impide llevar a cabo el sprint tal y como ha sido planificado. Generalmente, esta situación implica la reducción del número de puntos de historia que es capaz de llevar a cabo el equipo en el sprint o se limita alguna de las tareas necesarias para implementar las historias de usuario. El equipo, considerando la nueva situación, debe replanificar el sprint de forma que se siga entregando el mayor valor al cliente.
5. En la siguiente iteración, el Project owner entrega una nueva lista de historias de usuario al equipo. En esta nueva lista se pueden incluir nuevas historias de usuario o modificar/eliminar historias previamente definidas.
6. El equipo, considerando la nueva lista de historias de usuario, replanifica el product backlog. Para ello, estima el esfuerzo de las nuevas historias. Recalcula el ROI y la prioridad de las diferentes historias y genera un nuevo plan de proyecto.
7. Se vuelve al paso 3.

Cabe destacar que el número de iteraciones del proyecto está entre 6 y 8, dependiendo de cómo haya realizado el equipo la planificación del mismo. Sin embargo, a partir de la iteración 6 ya no hay modificaciones en la lista de historias de usuario ni situaciones que impidan llevar a cabo la iteración normalmente.

3.3 Scrum en Métodos de Producción de Software

Esta asignatura aplica Scrum dentro del Test-Driven Development (TDD) (Fucci & Turhan, 2014), que se basa en la construcción de tests unitarios y en la escritura del código necesario para que esos tests finalicen de forma satisfactoria. En este caso, Scrum se aplica en grupos de 3 alumnos. El profesorado ejerce el rol de Scrum máster y stakeholder, mientras que el alumnado juega el rol de desarrolladores y de product owner. Para alcanzar el objetivo O1, planteamos el aprendizaje basado en un proyecto (Morales & Landa,

2004). Durante todo el curso, los equipos de trabajo deben implementar un sistema software. En cada sesión se explicarán conceptos que permitirán al alumnado el desarrollo incremental del proyecto.

Para alcanzar el objetivo O2, las clases combinan la clase magistral con ejercicios prácticos. El trabajo autónomo se basa en el uso de 2 herramientas: GIT (<https://git-scm.com/>) y Jenkins (<https://www.jenkins.io/>). El uso de GIT les permite compartir el código software que van desarrollando de forma colaborativa y autónoma. En caso de conflicto entre versiones distintas del mismo software, GIT les permite resolver los conflictos, permitiendo al desarrollador elegir entre la versión que tiene en local o la versión que está guardada en el servidor. La herramienta Jenkins permite pasar los tests desarrollados por todo el equipo de forma automática. Esto permite garantizar que el código que ha escrito cada persona dentro del equipo, es compatible con el resto de tests que han hecho los otros miembros.

Para alcanzar el objetivo O3 se propone el uso de la herramienta Jira (<https://www.atlassian.com/es/software/jira>). Esta herramienta permite que los propios miembros del equipo se asignen tareas y responsabilidades. Esta asignación se hace conforme a las historias de usuario, priorizándola en base a la importancia que tiene esa funcionalidad para el sistema. El alumno responsable de cada historia de usuario debe hacer tanto los tests como el código funcional.

Para alcanzar el objetivo O4, se hace un seguimiento del alumnado a lo largo de todo el proceso de desarrollo. Se pide a cada equipo que defina el backlog y reparta las historias de este backlog en dos sprints. El profesorado debe hacer un seguimiento en cada sprint asegurando que la metodología de la Ingeniería del Software se aplica de forma correcta.

A continuación, se detalla la metodología docente de forma secuencial:

1. El profesorado propone un proyecto que es común para todos los equipos de desarrollo. Cada equipo de desarrollo define el product backlog en base a las historias de usuario.
2. Se hace una estimación temporal de cuánto tiempo les puede llevar a cabo el desarrollo de cada historia de usuario. Como estimador de coste de cada historia de usuario, se usa la técnica del planning póker (Mahnič & Hovelja, 2012).
3. Se dividen las historias de usuario en los dos sprints. Todos los miembros del equipo deben trabajar en los dos sprints.
4. En cada sprint se dedican 2 meses. Cada persona empieza elaborando los tests y escribiendo el código que hacen que el test funcione de forma correcta.
5. Al final de los dos sprints, cada equipo hace una presentación donde se valoran los 4 objetivos que se pretendían conseguir con Scrum.

4. Resultados

Esta sección va a mostrar las notas que cada equipo Scrum obtuvo en el curso 2021/2022 (el más reciente del que tenemos registro en las 3 asignaturas implicadas) en el proyecto desarrollado. Es importante destacar que estas notas no son las notas finales, pero el proyecto tiene un porcentaje elevado en la evaluación continua del 60%.

En el caso de Ingeniería del Software II, todos los alumnos aprobaron el proyecto de laboratorio en primera convocatoria. Sin embargo, en la nota final de la primera convocatoria sí que hubo algunos suspensos debido a que no aprobaron el examen final. La Fig 1 muestra los resultados de la nota de laboratorio de los 8 equipos.

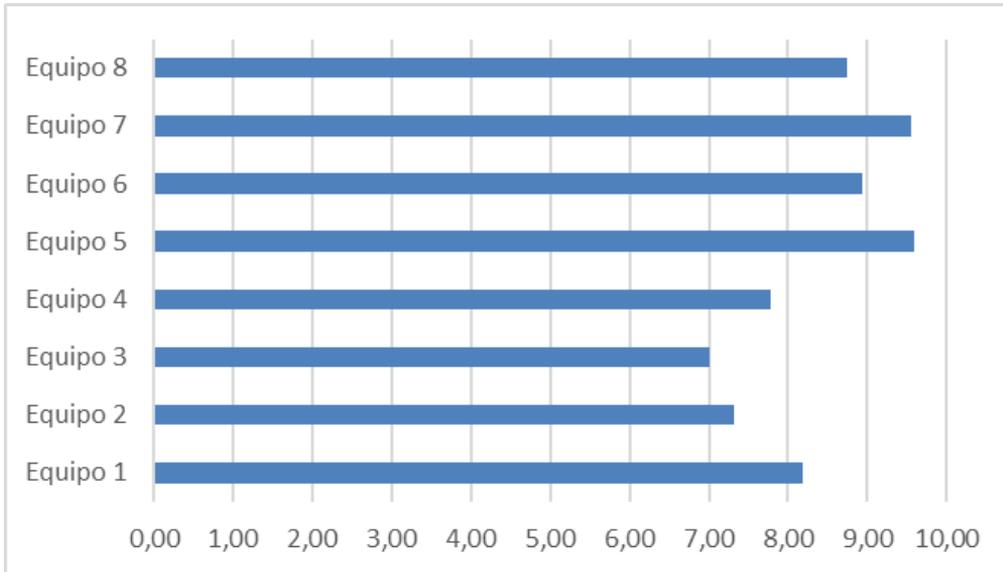


Fig 1. Notas de Ingeniería del Software II

En el caso de la asignatura de Gestión de Proyectos, todos los alumnos entregaron el proyecto y solo uno de los equipos suspendió. Sin embargo, la nota obtenida en el proyecto por este último equipo era suficiente para promediar con el resto de las notas de la asignatura, por lo que todos los miembros de dicho equipo aprobaron la asignatura en primera convocatoria. Del conjunto total de estudiantes de la asignatura solo uno suspendió la asignatura en primera convocatoria, debido a que no aprobó el examen final, y cinco personas no se presentaron al examen. El detalle de las notas se puede consultar en la Fig 2.

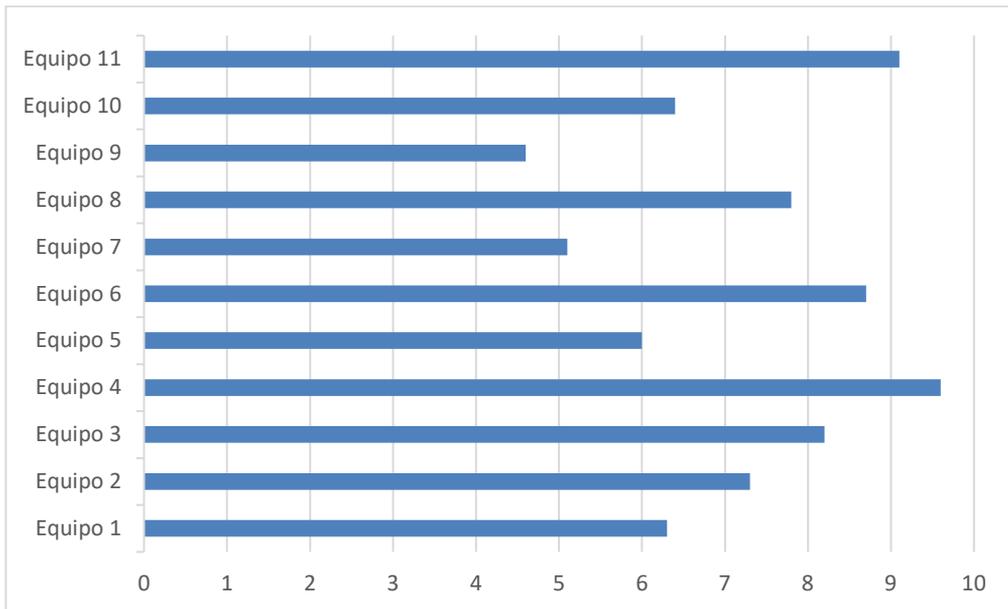


Fig 2. Notas de Gestión de Proyectos

En el caso de la asignatura de Métodos de Producción de Software, todos los alumnos entregaron el proyecto, y todos aprobaron en primera convocatoria. Cabe destacar que, al tratarse de un curso de máster, el alumnado está más motivado que en el grado y por tanto el porcentaje de entregados en primera

convocatoria suele ser del 100%. La Fig 3 muestra los resultados de los 6 equipos que participaron en el curso 2021/2022. Cabe destacar que todas las notas están entre 8,5 y 10, lo que demuestra que los objetivos docentes de Scrum se han adquirido de forma satisfactoria.

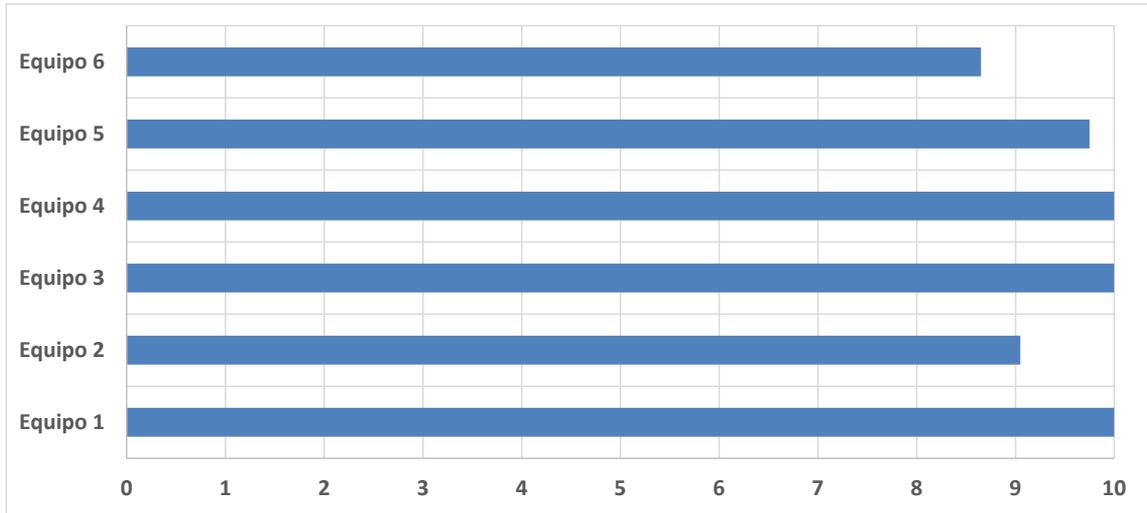


Fig 3. Notas de Métodos de Producción de Software

Para concluir, vamos a reportar algunas de las lecciones aprendidas en los 3 escenarios de uso de Scrum:

- El alumnado es muy ambicioso a la hora de asignar historias de usuario en el primer sprint. Creen que van a tener más capacidad de desarrollo de la que realmente tienen. Este error se corrige en los sucesivos sprints.
- Hay conflictos en algunos equipos de trabajo. No todos los miembros dedican el mismo esfuerzo ni están igual de motivados, lo que provoca a veces cierto mal estar en el equipo.
- Al jugar el alumnado dos roles a la vez (producto owner y desarrollador) muchas veces no saben si la tarea que hacen pertenece a un rol u otro.

5. Conclusiones

Este trabajo ha presentado cómo se aplica Scrum en tres asignaturas distintas relacionadas con la Ingeniería del Software. Las asignaturas son: Ingeniería del Software II, Gestión de Proyectos (ambas de GII), y Métodos de Producción de Software (del máster TWCAM). En las 3 asignaturas se han compartido los mismos objetivos docentes, aunque cada una utiliza Scrum desde una perspectiva diferente. El artículo aborda las particularidades de cada una de las asignaturas.

Como trabajo futuro, planteamos extender el uso de Scrum a otras asignaturas tanto del GII como del máster TWCAM donde se haga un desarrollo en equipos. Queremos estudiar la opción de coordinar las asignaturas de Gestión de Proyectos con Ingeniería del Software II de forma que los alumnos de Gestión de Proyectos hagan de coordinadores de los alumnos de Ingeniería del Software II.

6. Referencias

Aguila, I. M. d., Guirado, R., & Miranda, C. M. (2023a). *Juego de las monedas*: <https://imaguila.github.io/RepositorioJuegosInSo/Monedas/Descripcion.html>.

- Aguila, I. M. d., Guirado, R., & Miranda, C. M. (2023b). *Juego de los valores*: <https://imaguila.github.io/RepositorioJuegosInSo/CartasValores/Descripcion.html>.
- Cockburn, A. (2006). *Agile software development: the cooperative game*. Pearson Education.
- Fucci, D., & Turhan, B. (2014). On the role of tests in test-driven development: a differentiated and partial replication. *Empirical software engineering*, 19(2), 277-302. <https://doi.org/10.1007/s10664-013-9259-7>
- Guarino de Vasconcelos, L., Oliveira, L., Guimarães, G., & Ayres, F. (2018). Gamification Applied in the Teaching of Agile Scrum Methodology. In (pp. 207-212). https://doi.org/10.1007/978-3-319-77028-4_30
- Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), 2086-2095.
- Morales, P., & Landa, V. (2004). *Aprendizaje basado en problemas*. *Theoria* (Vol. 13).
- Schwaber, K., & Sutherland, J. (2011). The scrum guide. *Scrum Alliance*, 21(1), 1-38.
- Varela, A. (2023). *Juego planning póker*: <https://agustinvarela.org/estimaciones-relativas-con-planning-poker/>.