



Información del Material educativo

Nombre del curso	Metodologías de Desarrollo Ágil
Programa	Tecnología en desarrollo de Software
Facultad	Facultad de Ingeniería y Ciencias Ambientales
Número de créditos	2
Idioma	Español
Palabras clave	Agilismo, metodologías, desarrollo
Escritor	Roberto Carlos Guevara Calume
Asesoría pedagógica	Unidad de Creación de Contenidos
Diagramación y diseño gráfico	Unidad de Creación de Contenidos
Versión	1
Año	2020
Diseño instruccional	Unidad de Creación de Contenidos Fundación Universitaria Católica del Norte

Competencia Global

Llevar a cabo técnicas y procedimientos para la gestión predictiva y ágil de proyectos de desarrollo de *software*, teniendo en cuenta las restricciones de un proyecto predictivo: planificación, ejecución, seguimiento y control, y cierre.

Estructura

Elemento de competencia 1: Aplicar diferentes estrategias y prácticas para implementar la metodología de programación extrema.

Temas
Tema 1: Metodologías tradicionales y metodologías ágiles
Tema 2: ¿Qué es metodología XP?
Tema 3: Fases de la metodología XP
Horas AD: 10
Horas TI: 22

AD: trabajo con acompañamiento docente. **TI:** trabajo independiente del estudiante.

Elemento de competencia 2: Aplicar la metodología *Scrum* en un proyecto de desarrollo de *software*, siguiendo el ciclo de vida interactivo e incremental y cumpliendo con las revisiones durante el proceso de construcción.

Temas
Tema 1: <i>Scrum: Sprint</i> , estructura y roles
Tema 2: El proceso <i>scrum</i>
Tema 3: <i>Scrum vs XP</i>

Horas AD: 10

Horas TI: 22

AD: trabajo con acompañamiento docente. **TI:** trabajo independiente del estudiante.

Elemento de competencia 3: Regular el flujo de avance continuo de un proyecto, mediante la aplicación de *Kanban* como metodología de desarrollo ágil.

Temas
Tema 1: Método <i>Kanban</i> : reglas y pasos
Tema 2: Tarjetas <i>Kanban</i>
Tema 3: Comparación de metodologías: <i>XP</i> , <i>Scrum</i> y <i>Kanban</i>
Horas AD: 10
Horas TI: 22

AD: trabajo con acompañamiento docente. **TI:** trabajo independiente del estudiante.

Ruta de aprendizaje

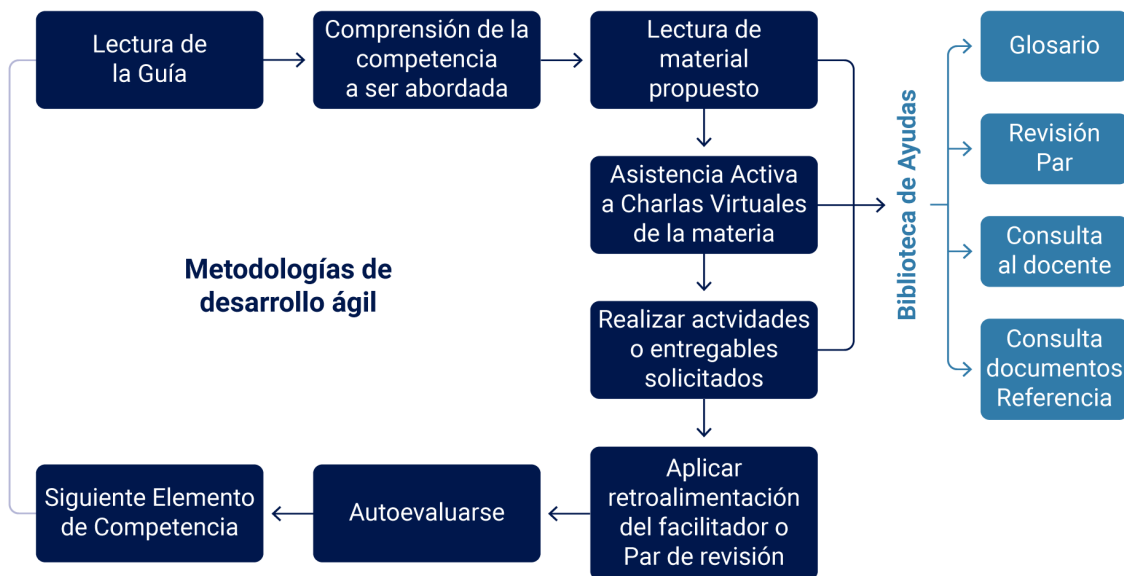


Figura 1. Ruta de aprendizaje. Elaboración propia (2019).

Descripción de la imagen: la imagen contiene como ruta de aprendizaje la guía de lectura, la comprensión de la competencia a ser abordada, la lectura de material propuesto, la asistencia activa a charlas virtuales de la materia, la realización de actividades o entregables solicitados, la aplicación de la retroalimentación del facilitador, la autoevaluación y el siguiente elemento de competencia; finalmente, como ayudas didácticas se encuentra el glosario, revisión de par, consulta al docente y consulta de documentos de referencia.

Introducción

Las metodologías ágiles tratan de hacer cambios con rapidez, según las demandas de cualquier tipo de proyecto, en cualquier tipo de empresa o proyectos de desarrollo de *software*. Estas metodologías son de gran ayuda cuando existen modificaciones al proyecto original, tiempo después de su inicio. En los años noventa surgieron metodologías de desarrollo de software ligeras –luego llamadas ágiles– dirigidas a reducir la probabilidad de fracaso por subestimación de costos, tiempos y

funcionalidades, en los proyectos de desarrollo de *software* (Navarro et al., 2016).

Las metodologías ágiles se empiezan a usar desde los años 80, para poder gestionar de forma ágil los proyectos en grandes empresas. Alrededor del año 2000 se creó una alianza ágil para el manejo de proyectos, y dichas metodologías se convierten en una serie de métodos que se aplican en este caso al desarrollo de *software*. Así, las metodologías ágiles han ganado popularidad desde hace algunos años, ya que constituyen una buena solución para proyectos a corto plazo, en especial, aquellos proyectos en donde los requisitos están cambiando constantemente (Amaya, 2013).

Las metodologías más relevantes son *Adaptive Software Development*, *Agile Unified Process (AUP)*, *Crystal Clear*, *Essential Unified Process*, *Feature Driven Development*, *Lean Software Development*, *Kanban*, *Open Unified Process (OpenUP)*, Programación Extrema, Método de desarrollo de sistemas dinámicos, *Scrum*.

Justificación

Para los profesionales en el desarrollo de *software*, es necesario conocer y aplicar las metodologías ágiles puesto que estas aumentan la productividad, permite actuar de forma interdisciplinaria, así cada integrante se puede enfocar en atención a su disciplina.

Las metodologías ágiles permiten al profesional en las TIC hacer una mejor gestión del riesgo de sus proyectos, puesto que permiten el desarrollo por ciclos continuos, entregas parciales y reajustar permanentemente los proyectos según los cambios o las necesidades que surjan. Sin embargo, las metodologías ágiles no evitan fallos o contratiempos, solo permiten poder enfrentar estos fallos de forma ágil y oportuna.

Según IEBS, la Escuela de Negocios de la Innovación y los Emprendedores "En esencia, las empresas que apuestan por las metodologías ágiles consiguen gestionar sus proyectos de forma flexible, autónoma y eficaz reduciendo los costes e incrementando su productividad" (IEBS, 2020).

En el ámbito académico y empresarial cada metodología ágil tiene ventajas que se pueden aprovechar y desventajas por lo que es necesario que los estudiantes puedan reconocer que metodología aplicar en Cada caso (Molina et al., 2018).

Metodología

Este curso es congruente con el modelo educativo que promueve el aprendizaje significativo, investigativo y cooperativo de forma autónoma y grupal, den el cual interactúan constantemente los diferentes actores que intervienen en la construcción de conocimiento para evidenciar las competencias requeridas.

La interacción es el aspecto central de toda experiencia educativa, muy especialmente bajo ambientes virtuales, sobre todo cuando se intenta promover el desarrollo del pensamiento crítico y reflexivo, y ocurre en el momento en que se establecen intercambios comunicativos entre el facilitador y el estudiante o entre los estudiantes; y también cuando el estudiante se enfrenta al análisis, lectura y retroalimentación de los materiales del curso y las actividades propuestas.

Las actividades de formación realizadas por el facilitador se diseñan partiendo de los resultados de aprendizaje derivados de los criterios de desempeño del elemento de competencia que se trabaja en la unidad.

- **Aprendizaje basado en problemas:** la premisa fundamental en el aprendizaje basado en problemas es que la enseñanza debe iniciarse con el planteamiento de problemas que sean de importancia y pertinencia para los estudiantes. Una teoría de diseño de este tipo de aprendizaje debe tener orientaciones que ayuden a secuencias de problemas contruidos unos sobre otros, en orden de complejidad. Cuando un problema finalmente se resuelve, los estudiantes aprenden volviendo al ámbito del problema, para replantearlo, evaluar su solución, recorrer las secuencias; en general, se determina un proceso de razonamiento posterior altamente constructivo.
- **Aprendizaje con gradualidad analógica:** desde un dominio de conocimiento fuente, hacia un dominio de conocimiento meta, que transfiere, rescatando y extrapolando con movimientos de

razonamiento analógico, los recursos del conocimiento fuente que ayudarán a constituir el conocimiento meta, posibilitando un proceso y ascenso gradual que da pasos cada vez más lejos y cada vez más densos y estructurados, en la permanente adquisición, consolidación y producción del conocimiento.

- **El proceso en las asignaturas se inicia con la entrega de un problema aplicativo:** la temática de la asignatura se irá desarrollando alrededor de este problema.
- El estudiante investiga conceptos relacionados con la asignatura en forma gradual, de acuerdo con las orientaciones del facilitador del curso.
- **Comprensión del contexto significativo:** mediante actividades de observación, descripción, análisis e interpretación en el contexto próximo del estudiante, a partir de problemas que se pueden resolver desde los procesos de investigación.
- **E-vida –Equipos virtuales de desarrollo académico–:** el aprendizaje colaborativo se integra con los procesos de comunicación en el ambiente virtual de aprendizaje, con la participación del facilitador y los estudiantes, bajo la premisa de trabajo colaborativo y construcción de comunidades de aprendizaje. Este trabajo colaborativo implica:
 - Creación de foros de debate para cada grupo, que permita la interacción con el facilitador y los compañeros de grupo.
 - Actividades en las evaluaciones que involucren trabajos en equipo e investigaciones con el grupo de trabajo.
 - Encuentros sincrónicos que permitan la resolución de inquietudes y el trabajo colaborativo y cooperativo.

Evaluación

La evaluación de las asignaturas se propone de la siguiente manera:

- **Portafolio:** (se detalla en la planeación del curso).

- **Examen tipo ECAES:** con preguntas de selección múltiple contextualizadas con las competencias que el estudiante puede desarrollar.
- Trabajo de aplicación.
- Desarrollo de actividades, talleres y ejercicios que se proponen en los foros.

Glosario

- **Adaptación:** cabe la posibilidad de que el objetivo cambie, o bien el modo en que lo alcanzamos. Hay que estar abiertos a cambiar de dirección y a la mejora continua (Hablemos de empresas, 2020).
- **Estrella de mar:** gráfico visual para mejorar la eficiencia de los equipos, mostrando en qué deberían focalizarse.
- **Equipo de desarrollo:** en este radica el éxito del proyecto. Quienes lo conforman adquieren un compromiso, deciden si se puede o no hacer lo que pretende el cliente en cada ciclo de desarrollo. Se comprometen a terminar la solución (Scrummanage, 2020).
- **Historia de usuario:** habla de lo que busca que haga el producto, presentado con una estructura más amigable para el desarrollador y para el usuario (Tenstep.ec, 2020).
- **Proceso predictivo:** es el típico de cascada en el cual suponemos que todo se hace una única vez. Se orienta con un plan, los requerimientos están fijos y se procura estimar el tiempo y los recursos necesarios (Searchdatacenter techtarget, 2020).
- **Procesos adaptativos:** funcionan con diferentes metodologías y estrategias de gerencia de proyectos. Scrum, uno de ellos, es un modelo que define ciertos papeles desempeñados en unas etapas durante la construcción. Los procesos adaptativos se

orientan a generarle valor al cliente de forma rápida (Bohorquez, 2013).

- **Product owner:** es la persona que vela por los intereses del cliente, particularmente en Scrum. Trabaja con el grupo de desarrollo y es responsable de priorizar los requerimientos y de darle valor a la entrega (Eude, 2020).
- **Product backlog:** es el conjunto de requerimientos del cliente y está integrado por historias de usuario. El product owner los selecciona, los prioriza, los mantiene, los administra, es quien sabe para dónde debería ir el desarrollo del producto y lo que más valor le da al cliente. El Scrum master y el equipo de desarrolladores son usuarios de product backlog (Programacionymas, 2020).
- **Sprint backlog:** a partir de la selección de un conjunto de los requerimientos del product backlog –realizada de común acuerdo con el cliente según el mayor valor para el negocio– se define el alcance de una iteración o un ciclo en el contexto de Scrum. Esto quiere decir que en el tiempo del sprint se van a desarrollar unos requerimientos funcionales escogidos de ese product backlog. Se implementa en un período sugerido de dos a cuatro semanas que, al finalizar, busca entregar valor, un producto rápido: el que se comprometió a desarrollar el equipo de trabajo (Bohorquez, 2013).
- **Punto de historia:** es una técnica de estimación y pretende darles un orden a las historias de usuario. Normalmente se suelen asignar valores, números enteros asociados a la serie de Fibonacci. Una historia entre 0 y 8 es manejable y si está entre 13 y 20 o 40 es muy grande (Hablemos de empresas, 2020).
- **Scrum master:** es un facilitador, ayuda al equipo de desarrollo para que el proceso se cumpla y colabora en la solución de problemas (Scrummanage, 2020).
- **Velocidad:** es el total de los puntos de historia que se es capaz de implementar en un sprint. Eso determina lo que el equipo de

desarrollo puede trabajar en una velocidad constante, estable (Scrummanage, 2020).

- **Iteración:** es un ciclo de tiempo que se va repitiendo. Tras cada iteración, el producto o servicio gana valor. Este se revisa hasta que el cliente queda conforme, por lo que es frecuente facturar por iteración (Searchdatacenter techtarget, 2020).

Referencias

Amaya, Y. D. (2013). *Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles*. Estado actual. Revista tecnologica, 2(2). www.doi.org

Bohorquez, I. (2013). *La organización empresarial como sistema adaptativo complejo*. Estudios Gerenciales, 29 (127) 258-265. www.scielo.org.co

Eude. (2018). *¿Cuáles son las responsabilidades de un Product Owner?* eude.es. www.eude.es

Martínez, M. (2018). *Conceptos básicos para que sepas qué es 'Agile'*. www.hablemosdeempresas.com

Molina, B., Vite, H., y Davila, J. (2018). *Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software*. Espirales Revista multidisciplinaria de investigación, 2 (269) 114-121. www.researchgate.net

Navarro, A., Fernandez, J., & Morales, J. (2016). *A review of agile methodologies for software development*. Prospect, 11(2), 30-39.

Programacionymas PYM. (2020). *Scrum: ¿Qué es el Product Backlog?* www.programacionymas.com

Scrum Manage. (2016). *Equipo de desarrollo* www.scrummanager.net

Searchdatacenter techtarget. (2017). *Analítica predictiva o análisis predictivo* www.searchdatacenter.techtarget.com

Tenstep.ec. (2016). *SCRUM: Cómo escribir historias de usuarios sin morir en el intento* www.tenstep.ec

Bibliografía complementaria

Canós, J., Letelier, P. y Penadés, M. (2003). *Metodologías ágiles para el desarrollo de Software: eXtreme Programming (XP)*, Universidad Politécnica de Valencia. www.cyta.com.ar

DSDM Consortium (2001). *Manifiesto por el Desarrollo Ágil de Software*. www.agilemanifesto.org

Ríos, E. y Wilson, S. (2008). *Desarrollo de un sistema informático para los procesos de cosecha y post cosecha de la camaronera Pampas de Cayanca*. (Tesis Pregrado). Facultad de ingeniería de sistemas, Escuela Politécnica Nacional, Quito. www.bibdigital.epn.edu.ec

Shenone M., H. (2013). *Diseño de una metodología Ágil de Desarrollo de Software*. (Tesis de pregrado). Universidad de Buenos Aires. www.materias.fi.uba.ar

Jyothi, V. E. y Nageswara Rao, K. (2012). «*Effective Implementation of Agile Practices - Incoordination with Lean Kanban.*». *International Journal on Computer Science & Engineering*, 4(1), 87-91.