

<b>Nombre del curso</b>	Taller de metodologías ágiles de desarrollo de software
<b>Descripción del curso</b>	<p>Los ingenieros de software estamos enfrentados a una actividad laboral eminentemente heurística, es decir, en donde todos los días se están resolviendo problemas distintos a los que se ha enfrentado anteriormente. De hecho, esto define a un buen ingeniero de software: el automatizar sistemáticamente procesos que son similares a la experiencia anterior.</p> <p>Por ende, un ingeniero de software está llamado a estar permanentemente navegando en aguas desconocidas, y su mejor herramienta para una productividad sostenible es el avanzar de manera sistemática creando colaborativamente (con sus clientes, usuarios, profesionales de especialidades complementarias y colegas) nuevo conocimiento a la vez que se generan nuevas soluciones informáticas. Esto es lo que da origen a la Agilidad.</p> <p>La industria del software está dominada por un paradigma tradicional en donde se asimila el desarrollo de software a un trabajo repetitivo tradicional, sin tomar en cuenta la incertidumbre inherente a esta labor</p>
<b>Objetivos</b>	En este taller los alumnos y alumnas aprenderán a detectar los problemas que el paradigma tradicional ha generado en la industria del software, e investigando que soluciones metodológicas han sido desarrolladas al alero de las metodologías ágiles, además de desarrollar contenido (artículos, videos, etc.) donde se capture lo aprendido. Muchas de estas soluciones son de reciente aparición en la industria (no más de 1 o 2 años) por lo que se requiere la capacidad de entender este contenido en el idioma de origen, esencialmente inglés.
<b>Contenidos</b>	<ul style="list-style-type: none"> <li>• Entendimiento Compartido y Diseño Colaborativo a través de Pensamiento Visual usando baja tecnología</li> <li>• Gestión de Flujo de trabajo usando Kanban</li> <li>• Resolución de problemas de negocio complejos de forma incremental usando Scrum</li> <li>• Desarrollo incremental de software con calidad intrínseca usando Extreme Programming</li> <li>• Validación de Innovaciones usando Lean Startup</li> <li>• Y nuevas metodologías que ocupen otros espacios de trabajo de la ingeniería de software</li> </ul>
<b>Modalidad de evaluación</b>	<ul style="list-style-type: none"> <li>• Proceso (30%) <ul style="list-style-type: none"> <li>○ reflexión y auto evaluación de como en este proyecto se pudo o no aplicar los principios ágiles que se han descubierto en el taller</li> </ul> </li> <li>• Producto (40%) <ul style="list-style-type: none"> <li>○ Periódicamente los alumnos deberán presentar sus avances específicos en la evolución del producto, los que serán evaluados y</li> </ul> </li> </ul>

	<p>retroalimentados por el equipo docente</p> <ul style="list-style-type: none"> <li>• Trabajo en Equipo <ul style="list-style-type: none"> <li>○ Coevaluación entre los miembros del grupo en relación a los aportes de cada integrante con respecto a los aportes: <ul style="list-style-type: none"> <li>▪ Entendimiento del Problema de Negocio a resolver</li> <li>▪ Trabajo en Equipo</li> <li>▪ Aporte Técnico</li> </ul> </li> </ul> </li> </ul>
<b>Bibliografía</b>	<p><b>Básica:</b></p> <ul style="list-style-type: none"> <li>• Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers, <ul style="list-style-type: none"> <li>○ Dave Gray, Sunni Brown, James Macanufo</li> <li>○ 2010</li> </ul> </li> <li>• Kanban: Successful Evolutionary Change for Your Technology Business <ul style="list-style-type: none"> <li>○ David J. Anderson</li> <li>○ 2009</li> </ul> </li> <li>• Extreme Programming Explained, 1st edition <ul style="list-style-type: none"> <li>○ Kent Beck</li> <li>○ 1999</li> </ul> </li> </ul>
	<p><b>Recomendada:</b></p> <ul style="list-style-type: none"> <li>• Visual Meetings: How Graphics, Sticky Notes and Idea Mapping Can Transform Group Productivity <ul style="list-style-type: none"> <li>○ David Sibbet</li> <li>○ 2010</li> </ul> </li> <li>• Innovation Games: Creating Breakthrough Products Through Collaborative Play <ul style="list-style-type: none"> <li>○ Luke Hohmann</li> <li>○ 2006</li> </ul> </li> <li>• Lean Software Development: An Agile Toolkit <ul style="list-style-type: none"> <li>○ Mary y Tom Poppendieck</li> <li>○ 2003</li> </ul> </li> <li>• Personal Kanban: Mapping Work   Navigating Life <ul style="list-style-type: none"> <li>○ Jim Benson, Tonianne de Maria</li> <li>○ 2011</li> </ul> </li> <li>• The People's Scrum: Agile Ideas for Revolutionary Transformation <ul style="list-style-type: none"> <li>○ Tobias Mayer</li> <li>○ 2013</li> </ul> </li> <li>• Clean Code: A Handbook of Agile Software Craftsmanship <ul style="list-style-type: none"> <li>○ Robert C. Martin</li> <li>○ 2008</li> </ul> </li> <li>• The Clean Coder: A Code of Conduct for Professional Programmers <ul style="list-style-type: none"> <li>○ Robert C. Martin</li> <li>○ 2011</li> </ul> </li> <li>• Growing Object-Oriented Software, Guided by Tests <ul style="list-style-type: none"> <li>○ Steve Freeman, Nat Pryce</li> </ul> </li> </ul>

	<ul style="list-style-type: none"><li>○ 2009</li><li>● Continuous delivery and DevOps: A Quickstart Guide<ul style="list-style-type: none"><li>○ Paul Swartout</li><li>○ 2014</li></ul></li><li>● Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing<ul style="list-style-type: none"><li>○ Gojko Adzic</li><li>○ 2009</li></ul></li></ul>
--	---