

**METODOS AGILES PARA
APLICACIONES WEB**

Año 2019

Carrera/ Plan:

Licenciatura en Informática Plan 2015/Plan 2012/Plan 2003-07
Licenciatura en Sistemas Plan 2015/Plan 2012/Plan 2003-07
Analista en TIC Plan 2017

Año: 2019**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Optativa**Correlativas:**

Orientación a Objetos 1

Ingeniería de Software 1

Profesor/es: Alejandra Garrido**Hs. semanales:** 6 hs**FUNDAMENTACIÓN**

Esta materia les permite a los alumnos profundizar los conocimientos adquiridos en el Paradigma de Orientación a Objetos y en conceptos básicos de Ingeniería de Software, en el contexto de la construcción de aplicaciones complejas como es el caso de las aplicaciones web con procesos de negocio.

En particular abarcaremos el desarrollo y el mantenimiento de aplicaciones web con metodologías ágiles, enfatizando los valores y principios de las mismas y ejercitando sus prácticas. Nos enfocaremos además en el desarrollo de software de calidad y centrado en el usuario, integrando métodos de evaluación y mejora de la usabilidad con prácticas ágiles.

Dentro de los métodos y técnicas que se abordarán para permitir construir y mantener la calidad del software se cubrirán temas de testing de distintos niveles y refactoring, tanto en relación a la calidad interna del código como a la calidad en cuanto a experiencia del usuario. También cubriremos conceptos y técnicas relacionadas al desarrollo ágil que han surgido con énfasis en los últimos años como el concepto de deuda técnica y las técnicas de visual management.

A través de esta materia los alumnos se expondrán al desarrollando un sistema real en un grupo de trabajo, y utilizando una metodología ágil muy actual en el ámbito profesional.

OBJETIVOS GENERALES

- Presentar a los alumnos la problemática de construir y mantener una aplicación compleja como es el caso de las aplicaciones web con procesos de negocio, teniendo en cuenta las necesidades de los usuarios, y respetando criterios de calidad interna y externa.
- Discutir los valores, principios y prácticas de las metodologías ágiles para la construcción de aplicaciones complejas en el contexto de un equipo de desarrollo, así como los principios y prácticas de métodos más específicos surgidos del movimiento ágil, como son “lean software development” y “visual management” de proyectos ágiles.
- Introducir nociones básicas de usabilidad, accesibilidad, experiencia del usuario y diseño de interacciones, así como los problemas comunes que surgen en la interacción del usuario con interfaces web y las posibles soluciones.
- Ejercitar el diseño centrado en el usuario a través del maquetado de interfaces.
- Exponer el concepto de “technical debt” en el mantenimiento de software y presentar técnicas específicas para su abordaje a través de testing y refactoring. Profundizar estas técnicas no solo para la mejora de atributos de calidad interna, sino también para aquellos de calidad externa como la usabilidad.

COMPETENCIAS

- LI-CE4- Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

-LS-CE1- Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

CONTENIDOS MINIMOS

- Valores, principios y prácticas de las metodologías ágiles. Características particulares de eXtreme Programming, Scrum y Lean software development.
- Principios y herramientas de Visual Management en proyectos ágiles.
- Arquitecturas de software. Arquitecturas web.
- Diseño Centrado en el Usuario. Usabilidad, Accesibilidad y Experiencia del Usuario. Maquetado de interfaces.
- Mantenimiento de software. Introducción al concepto de Technical Debt o Deuda Técnica. Tipos de Deuda Técnica. Abordaje de la Deuda Técnica.
- Testing. Tipos de tests: de aceptación, de unidad, de integración. Integración continua. Refactoring. Concepto de “bad smell”.
- Refactoring como técnica para mejorar la usabilidad de aplicaciones web. Concepto de “usability smell”. Refactoring de usabilidad en el cliente.

PROGRAMA ANALÍTICO

I - Metodologías ágiles de desarrollo y prácticas relacionadas

1. Metodologías Ágiles. Filosofía. Características fundamentales. Diferencias con las metodologías tradicionales de desarrollo de software.
2. Extreme Programming. Valores, principios y prácticas.
3. Scrum. Roles, reuniones y artefactos. Estimación de tareas.
4. Lean software development. Principios y prácticas Lean.
5. Visual management. Herramientas de visual management más reconocidas. Trello.

II - Arquitecturas de software

6. Definición de arquitectura de software. Diferentes abordajes.
7. Arquitecturas de software más reconocidas y patrones arquitecturales.
8. Arquitecturas web. Model-View-Controller en aplicaciones web.

III - Diseño centrado en el usuario y experiencia del usuario

9. Características del Diseño Centrado en el Usuario (UCD). Diferentes propuestas de integración de UCD con prácticas ágiles.
10. Nociones básicas de usabilidad y diseño de interacciones. Concepto de Experiencia del Usuario (UX). Accesibilidad en aplicaciones web.
11. Métodos de evaluación de la usabilidad. Heurísticas de Nielsen. Recorridos cognitivos. Guías de accesibilidad de la W3C. User Testing. Integración de user testing en un desarrollo ágil. A/B testing.
12. Maquetado de interfaces gráficas. Desarrollo de aplicaciones web con Mockup-Driven Development. Consideraciones de usabilidad y diseño de interacciones durante el maquetado.

III - Construcción y mantenimiento de software con métodos ágiles

13. Etapa de mantenimiento de software. Costo del mantenimiento
14. Technical debt o Deuda Técnica. Tipos de Deuda Técnica. Abordaje de la Deuda Técnica
15. Integración continua. Herramientas.
16. Refactoring. Características. Proceso de refactoring. El concepto de bad smell. Límites en la preservación del comportamiento. Herramientas. Refactoring en distintos lenguajes y paradigmas de programación. Refactoring de modelos y de otros artefactos de software. Refactoring de arquitecturas. Refactoring to patterns. Refactoring para la mejora de atributos de calidad internos (flexibilidad, mantenibilidad).
17. Test Driven Development. Características. Tipos de Testing: de unidad, de integración, de aceptación, de regresión. Características de cada uno. Estrategias de testing de unidad. Framework de testing de unidad: familia XUnit.

IV - Refactoring de usabilidad en aplicaciones web

18. Refactoring como técnica para mejorar la usabilidad y accesibilidad de aplicaciones web. Definiciones. Refactoring de usabilidad a nivel de modelos. Catálogo.
19. Malos olores de usabilidad o “usability smells”. Definición y catálogo.
20. Identificación automática de usability smells a partir de eventos de interacción.
21. Refactoring de usabilidad en el cliente o Client-Side Web Refactoring (CSWR). Catálogo de CSWR. Aplicación.

BIBLIOGRAFÍA

- Extreme Programming Explained. Kent Beck. Addison Wesley. 2005.
- Essential Scrum: A Practical Guide to the Most Popular Agile Process. Kenneth Rubin. Addison-Wesley. 2012.
- Lean Software Development. Dasari. Ravi Kumar. The PROJECT PERFECT White Paper Collection.
- Visual Management – A General Overview. Algan Tezel, Lauri Koskela, Patricia Tzortzopoulos. 5th Int. Conference on Construction in the 21st Century (CITC-V). 2009.
- Software Architecture: Perspectives on an Emerging Discipline. Mary Shaw, David Garlan. Pearson. 1996.
- Pattern Oriented Software Architecture. A system of patterns. Buschmann, Meunier, Rohnert, Sommerlad and Stal. Wiley. 1996.
- The evaluation of accessibility, usability and user experience. Helen Petrie and Nigel Bevan. The Universal Access Handbook, C Stepanidis (ed), CRC Press, 2009.
- Don't make me think. Revisited: A Common Sense Approach to Web Usability. Steve Krug. New Riders Publishing. 2013.
- Agile Design Process with User-Centered Design and User Experience in Web Interfaces: A Systematic Literature Review. J. Giacomelli Beux, E. A. Bellej, L. A. Brock, A. C. Bertoletti De Marchi, C. A. Holbig. LATIN AMERICAN JOURNAL OF COMPUTING, VOL. 5, NO. 2, NOVEMBER 2018.
- The evolution of agile UXD. Tiago Silva Da Silvaa, Milene Selbach Silveirab, Frank Maurerc, Fábio Fagundes Silveira. Information and Software Technology 102 (2018).
- Refactoring: Improving the Design of Existing Code. Fowler, Martin. Addison-Wesley, 1999.
- Test Driven Development. Kent Beck. Addison-Wesley. 1999.
- Automatic detection of usability smells in web applications. Julián Grigera, Alejandra Garrido, José Matías Rivero, Gustavo Rossi. Int. J. Hum.-Comput. Stud. 97: 129-148 (2017).
- Usability Improvement Through A/B Testing and Refactoring. Firmenich, Garrido, Grigera, Rivero, Rossi. Software Quality Journal 2018. DOI:10.1007/s11219-018-9413-y

METODOLOGÍA DE ENSEÑANZA

Los conceptos teóricos son presentados en clases teóricas por el profesor, y se les ofrecerá material extra a los alumnos que permita profundizar los temas durante las clases teóricas, a partir de distintos artículos que disparen la inquietud de los alumnos y que motiven la participación en clase.

Durante las clases prácticas los alumnos irán desarrollando una aplicación web en grupo, siguiendo una metodología ágil, que constará de diferentes instancias de entrega. El proyecto que los alumnos deben desarrollar profundizará la tecnología de orientación a objetos, utilizando prácticas que aseguren la calidad interna y externa de la aplicación (como testing y refactoring). Para esto los alumnos utilizarán distintas herramientas de acceso libre o provistas por la cátedra. Se le asignará un peso importante a la usabilidad de la aplicación resultante.

EVALUACIÓN

Para aprobar la **cursada** de la materia se requiere aprobar todas y cada una de las instancias de entrega del proyecto donde se desarrollará una aplicación web en forma grupal. Estas entregas son progresivas e incrementales, de manera que se hará un seguimiento puntual de cada grupo, intentando asegurar además una carga balanceada de trabajo entre los miembros del grupo. Los grupos tendrán la oportunidad de corregir o completar las entregas que tengan algún comentario antes de la siguiente instancia.

Para aprobar el **final** la materia se requiere aprobar un examen sobre los temas vistos en las clases teóricas. Los alumnos que deseen promocionar tendrán la oportunidad de dar una clase oral sobre un tema a acordar con el profesor durante la cursada, y así evitar el examen final.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Contenidos/Actividades	Evaluaciones previstas
1	Presentación de la materia. La importancia de las metodologías ágiles. Requisitos de la materia. Forma de aprobación. Fechas importantes. Presentación de proyectos posibles.	
2	Metodologías ágiles. Manifiesto, XP, Lean, Scrum. Herramientas de visual management: Trello	Presentación del proyecto seleccionado
3	User Centered Design. Integración de UCD con métodos ágiles.	
4	Nociones básicas de usabilidad y diseño de interacciones.	Presentación del "product backlog" y planificación del 1er sprint
5	Maquetado de interfaces web. Mockup-Driven Development. Integración de mockups con métodos ágiles. Técnica de maquetado.	
6	Métodos de evaluación de la usabilidad. Heurísticas y recorridos cognitivos. Testing de usuario. Integración de tests de usuario en un desarrollo ágil.	Presentación de mockups del 1er sprint
7	Arquitecturas de software. Patrones arquitecturales. Arquitecturas web: MVC.	
8	Mantenimiento de software. Concepto de deuda técnica. Refactoring de código. Code smells.	Presentación de 1er release del proyecto
9	Testing. Test Driven Development. Framework XUnit. Testing de aplicaciones web. A/B testing. Herramienta para crear tests de usuario.	Planificación del 2do sprint
10	Refactoring para mejorar usabilidad. Usability smells. Accessibility smells. Proceso ágil de mejora de la usabilidad durante el desarrollo. Refactorings en el cliente.	Presentación de mockups del 2do sprint
11	Accesibilidad. W3C Guidelines: WCAG	
12	Identificación de usability smells a partir de eventos de interacción	Presentación de 2do release del proyecto
13	Herramientas para la integración práctica de la evaluación y mejora de la usabilidad en el desarrollo web con métodos ágiles	
14	Integración final de temas	Presentación del producto final



Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

- Docentes:
 - Dra. Alejandra Garrido: garrido@lifa.info.unlp.edu.ar
- Plataforma virtual:
 - <https://catedras.info.unlp.edu.ar> . Curso: “Metodos Ágiles para Aplicaciones Web”

Firma del/los profesor/es