ORDENANZA N° 330

REGLAMENTO DE LA CARRERA DE MAESTRÍA EN INGENIERIA DE SOFTWARE.

Modalidad a Distancia.

La carrera de Maestría en Ingeniería de Software en Modalidad a Distancia se rige por el Reglamento de Actividades de Posgrado de la Facultad de Informática de la UNLP y la Reglamentación específica del Sistema de Educación a Distancia del Postgrado de la Facultad de Informática (SIED). Este Anexo Reglamentario detalla los aspectos particulares de la carrera en cuestión, en cada uno de sus artículos.

Fundamentación de la Maestría en Ingeniería de Software

La fundamentación de esta carrera se basa en la destacada trayectoria con la que cuenta la modalidad presencial de la misma, que se dicta desde el año 1997 y se encuentra acreditada desde el año 1999. En el transcurso de estos años, la Maestría en Ingeniería de Software a formado tanto a docentes e investigadores universitarios como profesionales de la industria de software, cubriendo una gran demanda de toda la región. Asimismo, resulta evidente el crecimiento constante del área de la ingeniería de software en todo el mundo, lo cual genera la necesidad, por el lado de la industria, de profesionales con conocimientos avanzados y el grado de rigor académico que requieren los posgrados, y por el lado de la universidad, de avanzar en la investigación que permita optimizar técnicas, métodos y procesos de la ingeniería de software.

En este contexto, a lo largo de estos años la carrera ha recibido estudiantes de todo el país, he incluso de otros países de la región, y expresiones de interés por realizar la carrera por parte de docentes de universidades públicas y privadas que no pudieron concretarse por el requerimiento de presencialidad. La modalidad a distancia surge para satisfacer dicha demanda en forma adecuada, nutriéndose de la experiencia adquirida en la virtualidad causada por el aislamiento social preventivo y obligatorio.

Esta carrera articula con la Especialización en Ingeniería de Software y con el Doctorado en Ciencias Informáticas de la misma Unidad Académica. La carrera cuenta con una planificada vinculación entre los cursos del plan de estudios y las competencias a desarrollar en relación con los objetivos de la carrera, lo cual se establece en los programas de cada curso. Los contenidos y bibliografía de estos programas se actualizan periódicamente para atender los nuevos temas de estudio e investigación, de forma acorde al gran desarrollo que sostiene el área en los últimos años, tanto en innovación en la industria como en artículos de investigación. De estos artículos se nutren las actividades de investigación que están descriptas en los programas de los cursos de la carrera.

En el ámbito de la Unidad Académica se desarrollan numerosos proyectos de investigación y proyectos de transferencia vinculados con la temática de la carrera, que involucran a la mayoría de los docentes de la misma. Los resultados de los proyectos de investigación han sido muy relevantes y pertinentes, con publicaciones en congresos y revistas internacionales de alto impacto. Adicionalmente las actividades de transferencia han sido las de mayor envergadura en las relaciones Universidad-Medio Productivo en Argentina. Ambos tipos de actividades han impactado positivamente hacia una mejora sustancial del perfil de la carrera, el nivel científico de sus docentes, y la participación de alumnos en los proyectos de investigación a través de trabajos finales y de tesis.

Con el objetivo de facilitar el acceso a la carrera a estudiantes de todo el país y países de la región,

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

y cubriendo la demanda creciente en el área de la ingeniería de software tanto en el ámbito académico como productivo en forma más amplia, surge la modalidad a distancia de esta carrera, que combina encuentros sincrónicos y actividades mediadas por diferentes herramientas tecnológicas, para el seguimiento continuo de los estudiantes por parte de docentes y tutores.

Artículo 1. Objetivo y perfil del egresado

La Carrera de ***Maestría en Ingeniería de Software*** tiene entre sus objetivos específicos:

- Formar recursos humanos altamente capacitados para la construcción sistemática de artefactos de software y la producción de conocimientos científicos en el área de la Ingeniería del Software.
- Generar y mantener actividades de investigación, desarrollo y transferencia tecnológica en el área de la Ingeniería del Software.
- Contribuir a mejorar el proceso de construcción de productos de software en la industria mediante la transferencia de conocimientos en áreas específicas de la Ingeniería del Software.

En este contexto la Maestría en Ingeniería de Software tiene dos direcciones convergentes: por un lado, generar recursos humanos de alto nivel para realizar investigación en tópicos vinculados a la ingeniería del software; por otro lado y como consecuencia de estas actividades de investigación, formar recursos humanos con una alta capacitación profesional y que sean capaces de contribuir a la transformación de la industria informática y de la construcción de productos de software en el mercado. Al mismo tiempo se trata de formar graduados con capacidad de I+D+I que puedan completar el Doctorado en Ciencias Informáticas, continuando los ejes temáticos de la Maestría.

Para lograr estos objetivos, y acorde a la modalidad a distancia, se incentiva la realización de actividades en línea (consultas, participación en foros de discusión, autoevaluaciones), prácticas de investigación y revisión bibliográfica, y reflexión guiada sobre las temáticas que se abordan.

La Carrera de Maestría en Ingeniería de Software otorga el título de *Magíster en Ingeniería de Software*.

Artículo 2. Competencias del Egresado

El egresado de la Maestría en Ingeniería de Software de la Universidad Nacional de La Plata deberá ser capaz de construir productos de software usando técnicas de avanzada, producir nuevos modelos de proceso para la construcción de los mismos, así como de generar conocimientos en el área de la Ingeniería del Software.

Las competencias del egresado son las siguientes:

C.1- Manejar y aplicar tecnologías actuales para el desarrollo de sistemas de software, incluyendo métodos, lenguajes, arquitecturas, frameworks y herramientas.

C.2- Tener capacidad para analizar diferentes modelos de proceso de desarrollo de software y evaluar su calidad tanto en aspectos del producto resultante como en la gestión de los individuos involucrados y sus interacciones

C.3- Gestionar, planificar y controlar proyectos de software de distinta envergadura.

C.4- Definir parámetros de calidad tanto interna como externa de un producto software, y establecer procesos de evaluación y mejora que atiendan la satisfacción de todos los involucrados (el cliente, los usuarios y su experiencia, y el equipo de desarrollo).

C.5- Diseñar y gestionar el almacenamiento de datos, aplicando diferentes tecnologías y frameworks de persistencia de datos actuales.

C.6- Tener capacidad de analizar el estado del arte en los distintos aspectos de la ingeniería de software, así como producir conocimiento científico en el área.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Artículo 3. Modalidad educativa de la Carrera y modelo pedagógico

La carrera propone una modalidad educativa híbrida que combina la realización de encuentros presenciales y/o por videoconferencia y el trabajo mediado por tecnologías digitales, en particular por un entorno virtual de enseñanza y aprendizaje (EVEA). El diseño de los cursos debe incentivar la participación activa de los estudiantes en los encuentros sincrónicos y foros de discusión, así como las actividades de análisis y reflexión de bibliografía y contenidos digitales, actividades de investigación y de transferencia de conocimiento a los propios contextos de trabajo.

Los objetivos, contenidos y competencias abordadas en cada materia determinan, en cada caso, el desarrollo de los encuentros sincrónicos y el tipo de actividades que se proponen. Asimismo, en todos los cursos se debe realizar un encuentro inicial orientado a presentar la planificación del curso y sus contenidos, así como establecer las pautas de cursado y aprobación. Además, se debe realizar un encuentro de integración final y/o evaluación, orientado a compartir las producciones realizadas, debatir sobre la integración de los contenidos y competencias ganadas en el desarrollo del curso y/o realizar una evaluación sumativa. Ambos encuentros podrán realizarse de forma presencial y/o por videoconferencia.

Cada curso debe contar con una propuesta en la que se incluya:

- **Programa del curso**: donde deberán constar objetivos, competencias a desarrollar en relación con el objetivo de la carrera, contenidos mínimos, detalle de los contenidos, modalidad de evaluación y acreditación, recursos y materiales de estudio, actividades experimentales y de investigación planificadas para la apropiación de los saberes y la evaluación, y la bibliografía básica y complementaria. Las herramientas tecnológicas a utilizar se podrán seleccionar acorde a los recursos que dispone el SIED del Postgrado de la Facultad de Informática.
- **Cronograma**: el cronograma deberá describir la secuencia temporal de los contenidos y las actividades del curso. El cronograma podrá ser semanal o quincenal (acorde a lo descripto en la reglamentación del SIED del Postgrado de la Facultad de Informática).
- **Actividades**: cada actividad planificada deberá especificar su consigna, las herramientas tecnológicas a utilizar, organización de tutorías de consulta y criterios de aprobación.
- **Evaluación de la propuesta**: acorde a la reglamentación del SIED en que se inserta la carrera, se debe proponer la forma de evaluar la propuesta del curso (materiales de estudio, desempeño de los docentes y tutores, mediación tecnológica, la propuesta en general, las actividades, etc.). Los docentes podrán utilizar las encuestas modelos que provee el SIED.

Artículo 4. Estructura de gobierno

La carrera cuenta con un Director y un Comité Académico, cuyas características y funciones corresponden a las indicadas en el Art. 7 del Reglamento de Actividades de Postgrado de la Facultad de Informática.

El Director debe tener categoría de Profesor Titular con dedicación exclusiva y nivel mínimo de Investigador Científico sin Director, reconocida trayectoria académica y lugar de trabajo en la Facultad de Informática de la UNLP. El Comité Académico está integrado con investigadores del máximo nivel del país y del exterior designados por el HCD de la Facultad de Informática en base a sus antecedentes académicos. El Comité Académico se reunirá a solicitud del HCD por pedido del Director de Postgrado (que forma parte de la Coordinación del SIED), y estará encargado de realizar la evaluación externa de la carrera y de colaborar con el Director de la carrera en la organización de la misma y evaluar las propuestas de Tesis en los casos que se le solicite.

El Director de la carrera participa en la coordinación del SIED en los aspectos particulares referidos a la carrera y su modalidad, atendiendo a aspectos de calidad y de su seguimiento.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Artículo 5. Duración de la Carrera

El plazo estipulado para la realización de las actividades tendientes a obtener el Grado Académico de Magister en Ingeniería de Software no podrá ser menor a (2) dos años ni mayor a cinco (5) años, a partir de la fecha de inscripción.

Los cursos de la Maestría requieren 24 meses y los alumnos tendrá un plazo máximo de 12 meses a partir de la aprobación de los cursos para presentar y aprobar su Tesis de Maestría.

Eventualmente, el Honorable Consejo Directivo podrá conceder una prórroga a este plazo para la finalización de la Tesis de Maestría ante la solicitud fundamentada del aspirante. Esto requerirá mayoría especial de HCD (dos tercios del total de los miembros del cuerpo).

Artículo 6. Estructura de la Carrera. Carga Horaria

La Maestría en Ingeniería de Software es una carrera de tipo estructurado.

La carrera se estructura a partir de 7 cursos teórico-práctico obligatorios (5 de ellos destinados a abordar los saberes propios de la carrera, 2 vinculados al proceso de formación en tareas de investigación y redacción de publicaciones científicas y tesis, y finalmente, la realización de la Tesis de Maestría).

La realización de los Seminarios de Metodología de la Investigación y Taller de Tesis como prerrequisitos para la presentación de la Tesis, son comunes con las otras carreras de Maestría y se diferencian a través de un módulo ad-hoc para la carrera en cuestión.

Se proponen además actividades complementarias tales como cursos/seminarios optativos, trabajos, tutoriales y contenidos adicionales, que tienen como finalidad nivelar los conocimientos de base para el trabajo conjunto, u ofrecer alternativas para el acercamiento a líneas de I+D+I que pueden ser de interés para el desarrollo de la tesis. Los maestrandos pueden participar en tareas de investigación en el ámbito de la Facultad en vinculación a los temas de su tesis.

La evaluación de los cursos sigue el método que especifique el docente (en general proyectos individuales a presentarse en un plazo breve luego de dictado en curso). En todos los casos existe constancia escrita de la misma y el enfoque se orienta a la Investigación en los temas propios de la Maestría.

Cursos Obligatorios

Asignatura	**Carga horaria**	**Horas presenciales/ VC[1]**	**Horas no presenciales**	**Corr.**
1. Diseño de Bases de Datos	108	30	78	---
2. Tópicos de Ingeniería de Software I	108	50	58	---
3. Técnicas y Herramientas	108	30	78	---
4. Tópicos de Ingeniería de Software II	108	40	68	3
5. Administración de Proyectos	108	50	58	--
6. Metodología de la Investigación	70	25	45	--
7. Taller de Tesis	50	20	30	--
Tesis de Maestría	300			1,2,4,5, 6,7

Total de horas presenciales/VC: 245 hs
Total de horas no presenciales: 415 hs
Total de horas de investigación y Tesis de Maestría: 300 hs
TOTAL de Horas. 960 hs

[1] VideoConferencia

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

El trabajo de Tesis de Maestría debe reflejar un estudio detallado y actualizado del estado del conocimiento en el área específica del Ingeniería de Software y una investigación o desarrollo aplicado propio que constituya un aporte creativo a nivel nacional.

El trabajo de Tesis puede ser complementado con presentaciones a Congresos o publicaciones reconocidas sobre el mismo tema, de las que el Tesista sea autor o coautor.

Artículo 7. Modalidad de evaluación de los cursos y seguimiento de alumnos

Todos los cursos deberán involucrar un proceso de evaluación formativa y sumativa. También podrán incluir una evaluación diagnóstica para analizar los conocimientos previos de los alumnos. Para la evaluación final se utilizará una escala numérica, considerando un rango de 1 (uno) a 10 (diez), siendo 10 la máxima calificación alcanzable y 6 la mínima para aprobar.

Cada docente responsable deberá plantear el método de evaluación particular del curso (en general proyectos individuales a presentarse en un plazo breve luego de dictado del mismo) y el enfoque deberá orientarse a la investigación en los temas propios de la Maestría y sus objetivos. Los docentes pueden solicitar a los alumnos que las producciones sean compartidas con el resto de los compañeros, de manera tal que también se convierta en una instancia de aprendizaje (esto puede darse a través de un encuentro específico para este fin ya sea presencial o por VC, y pueden complementar con las herramientas disponibles en el EVEA como el repositorio compartido para publicar las producciones). Cada actividad que se proponga en el programa de una asignatura deberá indicar también los criterios de evaluación y seguimiento que se considerarán.

En todos los casos se dejará constancia escrita del resultado de la evaluación.

Los docentes y tutores que guían y acompañan el dictado, deberán realizar el seguimiento de los alumnos, analizando las dificultades que se presentan en referencia a los temas abordados y a su propuesta metodológica.

Por otra parte, la Secretaría de Postgrado, junto con la coordinación del SIED, deberá cumplir un rol participativo en la orientación de los alumnos en referencia a cuestiones administrativas de la carrera, y de seguimiento general, relacionadas con el registro de notas, gestión de expediente de cada alumno, registro de participación de los encuentros, entre otros. Se cuenta con un sistema informático para el almacenamiento y seguimiento de la información.

En cuanto al rol del Director de la Maestría en los procesos de seguimiento, éste deberá encargarse de supervisar todos estos procesos y se vinculará con la Coordinación del SIED para la mejora de la calidad de la carrera. Además, podrá realizar entrevistas informales (vía VC o presenciales) con los estudiantes con el fin de analizar el funcionamiento de la carrera en general. Asimismo deberá proponer actividades adicionales a los estudiantes para complementar su formación e información, así como orientarlos en la planificación de la propuesta de Tesis.

Artículo 8. Tareas experimentales a realizar en la carrera

La Maestría en Ingeniería de Software tiene un enfoque teórico-práctico en todos sus cursos, que incluye trabajos prácticos y proyectos de mediana embergadura relacionados con:

- Análisis y diseño de aplicaciones usando técnicas modernas
- Reuso y diseño de arquitecturas usando patrones y frameworks
- Modelización de requerimientos de software
- Diseño de bases de datos, especialmente de bases de datos no convencionales
- Métodos y herramientas de gestión y calidad de proyectos de software
- Métodos y técnicas de mejora de la calidad interna y externa de una aplicación relacionada a atributos no-funcionales.
- Aplicaciones de la Inteligencia Artificial a la Ingeniería de Software.
- Posibilidad de realizar Pasantías en el LIFIA en los temas de la Maestría.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Artículo 9. Alumnos. Inscripciones

a) Las inscripciones a la Maestría en Ingeniería de Software podrán efectivizarse al menos en dos períodos anuales, de acuerdo a lo que establezca el HCD de la Facultad. La inscripción la realiza el alumno en la Secretaría Administrativa de Postgrado.

b) La inscripción de los egresados con título máximo de grado de la Facultad de Informática de la UNLP, así como la de los egresados de carreras de Informática de título máximo de grado de las Universidades que forman la Red de Universidades Nacionales con Carreras en Informática (RedUNCI) será aceptada automáticamente.

c) También podrán inscribirse en la Carrera egresados con título universitario de otras Unidades Académicas de la Universidad Nacional de La Plata o de otras Universidades Nacionales o privadas, o de Instituciones acreditadas del extranjero que sean considerados equivalentes a los otorgados por la UNLP. En todos los casos deberán presentar Curriculum Vitae del postulante, incluyendo copia del título, certificado analítico de estudios, planes de estudio y programas detallados para la obtención del título de grado. En estos casos el Director de la Maestría y la Comisión Asesora de Investigaciones y Postgrado podrán fijar requerimientos (cursos / evaluaciones) previos a la aceptación de la inscripción.

d) En el caso de egresados terciarios con título específico en Informática y dilatada experiencia profesional en Informática, se podrá aceptar su inscripción previa evaluación de conocimientos por parte del Director de la Maestría y recomendación explícita de la Comisión Asesora de Investigaciones y Postgrado, debiendo tener mayoría especial en el HCD (2/3 del total de los miembros del cuerpo).

Artículo 10. Tesis de Maestría

a) El Trabajo de Tesis de Maestría en Ingeniería de Software deberá ser individual y exponer con claridad la tarea de investigación bibliográfica y estado del arte realizada y los aportes creativos (teóricos o de desarrollo) resultantes sobre el tema elegido.

b) Una vez aprobados los cursos regulares y realizado el taller de Redacción de Tesis y el Seminario de Metodología de la Investigación, el alumno de la Maestría elevará una propuesta de tema y el plan de Tesis y el Director / Codirector que lo avalan (según el formato establecido en el Art. 16). Esta propuesta, acompañada por los antecedentes del Director y del Codirector (si corresponde), será considerada en primer lugar por el Director de la Maestría y el Director de Postgrado. Si la propuesta se considera viable se elevará al HCD de la Facultad para su aprobación, vía la Comisión Asesora de Investigaciones y Postgrado. En caso contrario, propondrán al postulante la realización de modificaciones en la propuesta, o el rechazo definitivo de la misma.

Artículo 11. Dirección de las Tesis de Maestría

a) La dirección de la Tesis de Maestría en Ingeniería de Software podrá ser ejercida por un Director que podrá tener un Codirector o por dos Directores. Por razones de carácter extraordinario y con la debida fundamentación, la dirección de Tesis de Maestría en Ingeniería de Software podrá ser ejercida por un Director y dos Codirectores. Tanto Director/es como Codirector/es deben ser Profesores Universitarios del país o del exterior con méritos reconocidos en el área temática de la Tesis. Al menos uno de ellos deberá realizar tareas de Investigación y Desarrollo en el ámbito de la Facultad de Informática de la UNLP o dictar clases de grado o postgrado en esta Facultad. Director/es y Codirector/es podrán tener a su cargo un máximo de 5 tesistas cada uno incluyendo los de otras carreras de postgrado.
b) En el caso de Tesis de Maestría que se desarrollen en el marco de convenios de cooperación con Universidades del exterior, podrán tener hasta 2 Directores locales (o 1 Director y 1

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Codirector) por la Facultad de Informática de la UNLP que realicen Investigación en la misma y 2 Directores externos (o 1 Director y 1 Codirector), por la Universidad con la que hubiera Convenio.

c) En todos los casos Directores/CoDirectores deberán poseer una sólida versación en el tema de tesis propuesto y desempeñarse con independencia en la planificación y ejecución de actividades de investigación y desarrollo. Los antecedentes de Director/es / Codirector/es acompañarán la propuesta de Tesis de Maestría.

d) Los requisitos mínimos (alternativos) para ser Director de Tesis de Maestría son:
d.1- Tener título de Postgrado acreditado de Magister o Doctor, ser Profesor Universitario con al menos 3 años de antigüedad y tener antecedentes en formación de recursos humanos en el tema de Tesis. En caso de estar categorizado como docente-investigador tener al menos categoría III. En caso de no estar categorizado, la Comisión Asesora de Investigaciones y Postgrado evaluará la equivalencia con la categoría III.
d.2- Ser Profesor Titular o Asociado con al menos 5 años de antigüedad, estar categorizado como docente-investigador I, II o III y tener antecedentes en dirección/codirección de proyectos de investigación y desarrollo acreditados y en formación de recursos humanos en el tema de la Tesis.
d.3- Tener título de Postgrado acreditado de Magister o Doctor, pertenecer a la carrera del Investigador de CONICET o CIC, tener participación al menos en los últimos 5 años en proyectos acreditados de la Facultad de Informática de la UNLP y haber dictado cursos en el Postgrado de la Facultad al menos en los últimos 3 años.
d.4- Tener título de Postgrado de Doctor o Magister, ser Profesor ordinario de la Facultad de Informática de la UNLP al menos en los últimos 3 años, tener participación al menos en los últimos 5 años en proyectos acreditados de la Facultad de Informática de la UNLP y haber dictado cursos en el Postgrado de la Facultad al menos en los últimos 3 años.

e) Requisitos mínimos (alternativos) para ser Codirector de Tesis de Maestría:
e.1- Tener título de Postgrado acreditado de Magister o Doctor, ser Profesor Universitario con al menos 3 años de antigüedad y tener antecedentes en formación de recursos humanos en el tema de Tesis. En caso de estar categorizado como docente-investigador tener al menos categoría III. En caso de no estar categorizado, la Comisión Asesora de Investigaciones y Postgrado evaluará la equivalencia con la categoría III.
e.2- Ser Profesor Titular o Asociado con al menos 5 años de antigüedad, estar categorizado como docente-investigador I, II o III y tener antecedentes en dirección/codirección de proyectos de investigación y desarrollo acreditados y en formación de recursos humanos en el tema de la Tesis.
e.3- Tener título de Postgrado acreditado de Magister o Doctor, tener participación al menos en los últimos 3 años en proyectos acreditados de la Facultad de Informática de la UNLP y haber dictado cursos en el Postgrado de la Facultad al menos en los últimos 2 años.
e.4- Tener título de Postgrado acreditado de Magister o Doctor, ser Profesor ordinario de la Facultad de Informática de la UNLP al menos en los últimos 2 años, tener participación al menos en los últimos 4 años en proyectos acreditados de la Facultad de Informática de la UNLP y haber dictado cursos en el Postgrado de la Facultad al menos en los últimos 2 años.

f) Serán funciones del Director / Codirector de Tesis:

- Juntamente con el alumno, definir el tema de tesis y elaborar el respectivo plan de trabajo.
- Refrendar, cuando corresponda, las eventuales modificaciones en los planes de tesis
- Asesorar, dirigir y evaluar el desarrollo de las actividades del tesista.

g) Tanto Director/es como Codirector/es podrán renunciar a la dirección de la Tesis de Maestría, mediante una nota fundada dirigida al Director de la Maestría, quien resolverá cómo prosigue el trabajo de Tesis el alumno. También el alumno puede solicitar al Director de la Maestría algún cambio en la dirección de su Tesis de Maestría, lo que obligará a presentar nuevamente la propuesta de Tesis.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Artículo 12. Presentación de las Tesis de Maestría

Una vez completados los cursos y el trabajo de Tesis, el alumno con el aval de su Director elevará cinco (5) ejemplares impresos de la Tesis (según el formato establecido en el Art. 17), cinco (5) copias de la Tesis en soporte digital, y solicitará la constitución del Jurado que evaluará la misma. La tesis podrá ser presentada a partir de cumplidos seis meses de la aprobación del plan propuesto.

Tanto la escritura del trabajo de Tesis de Maestría en Ingeniería de Software como su defensa podrán ser realizados en lengua española o portuguesa. El Maestrando, con el aval de su Director y con causas debidamente justificadas, podrá solicitar redactar su Tesis en otro idioma. La solicitud deberá presentarse al menos 6 meses antes de la elevación de la Tesis para su análisis y será tratada por el Honorable Consejo Directivo de la Facultad de Informática como excepción. En caso de aprobarse la redacción de la Tesis en otro idioma, el Maestrando deberá entregar en castellano el Objetivo de la Tesis, el Detalle de su contenido, una Síntesis que incluya los Aportes y Resultados obtenidos y el análisis de las Conclusiones resultantes de la Tesis. Esta información se registrará como Anexo a la documentación en otro idioma.

A fin de apoyar la valoración de la Tesis de Maestría realizada, el alumno podrá acompañar las publicaciones que referidas al tema de la misma haya realizado durante su trabajo. En caso de no tener producción científica/profesional asociada con el trabajo de Tesis, tanto la Comisión Asesora de Investigaciones y Postgrado como el Jurado podrán exigirla como prerrequisito para evaluar la Tesis.

Artículo 13. Jurados de Tesis de Maestría y Evaluación de la Tesis.

a) La Comisión Asesora de Investigaciones y Postgrado propondrá al HCD la constitución de un Jurado encargado de evaluar la Tesis y la defensa oral y pública de la misma. Este Jurado estará integrado por tres (3) miembros titulares y un (1) miembro suplente que deberán ser Profesores Universitarios del país o del exterior de reconocido prestigio y conocimientos en el tema de la Tesis, siendo por lo menos 1 de ellos Profesor externo a la UNLP. El Director no participa del Jurado.

b) Una vez designado el Jurado, podrá ser recurrido por el alumno, mediante presentación fundada ante el HCD dentro de los 3 días hábiles siguientes a la designación. Esta recusación será tratada y resuelta con el asesoramiento de la Comisión de Investigaciones y Postgrado, siendo la resolución del HCD inapelable. Las causales de recusación serán las mismas que para los concursos de profesores ordinarios de la UNLP (de acuerdo a lo aprobado en el art 24 inc b) del Reglamento de Actividades de Postgrado de la Facultad de Informática).

c) Dentro de los 30 días de constituido el Jurado, éste deberá expedirse sobre la aceptación o rechazo del trabajo de Tesis y fijar fecha para la defensa pública del mismo. Cada uno de los Jurados deberá presentar una nota de evaluación que contenga su opinión sobre:

- Aporte de la Tesis presentada.
- Profundidad de la investigación/desarrollo realizado.
- Metodología del trabajo adoptada y aplicada.
- Calidad del trabajo experimental (si correspondiera).
- Claridad y precisión de la redacción.
- Fuentes de información y Bibliografía.
- Validez de las conclusiones alcanzadas.
- Concluirá la nota con una evaluación final sintética en la que indicará su aceptación o no para la defensa oral de la Tesis.

d) Para habilitar la defensa oral del Trabajo de Tesis se requerirá la opinión favorable de la mayoría de los miembros del Jurado.

e) En caso de estar habilitada para su exposición, se sugiere un plazo máximo de otros 30 días, siendo atribución de la Secretaría de Ciencia, Técnica a través de la Prosecretaría de Postgrado la coordinación de la exposición con el Jurado. Estos plazos podrán prorrogarse, por pedido de alguno

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

de los miembros del Jurado por un máximo de 30 días adicionales.

f) En caso de no aceptación para exposición, la opinión escrita de los Jurados (y sus indicaciones/sugerencias para el Tesista si las hubiere) será comunicada formalmente al alumno, a su Director y al Director de la Maestría correspondiente. En este caso, transcurridos 120 días el alumno podrá presentar por segunda vez su Trabajo de Tesis (con las correcciones que correspondieran). Si es nuevamente rechazado, no podrá volver a presentarlo y esta medida será inapelable.

Artículo 14. Defensa oral y pública de la Tesis.

a) Será obligatoria la Defensa Oral y Pública del Trabajo de Tesis. Este acto revestirá el carácter de Académico y deberá contar con la presencia de al menos dos (2) miembros del Jurado. Con anterioridad a la exposición, los miembros del Jurado podrán mantener una entrevista con el tesista en la que podrá estar presente el Director de Tesis.

b) El desarrollo del acto estará dirigido por un profesor designado por la Facultad de Informática. Este profesor dará por iniciado el acto, dirigirá el debate posterior, si lo hubiera, y dispondrá el orden en el cual el Tesista deberá contestar los diversos interrogantes que le planteen los miembros del Jurado. Cuando no hubiera más preguntas, dicho profesor dará por finalizada la defensa.

c) Finalizada la defensa oral y pública, se levantará un Acta de evaluación con la firma de los Jurados, el Tesista y el Director de Tesis. En el Acta el Jurado indicará la valoración de los puntos indicados en el Art. 13 de este anexo, la calidad de la exposición oral y los conocimientos demostrados en las respuestas a los interrogantes planteados a fin de establecer una calificación.

La calificación final podrá ser Excelente (10), Distinguido (9 u 8), Muy Bueno (7 o 6) o Insuficiente. Se entregará una copia del Acta al tesista, se anexará otra copia al expediente para realizar la comunicación al HCD, y en caso de ser aprobada se remitirá otra copia a la UNLP. Todas las decisiones del Jurado serán inapelables.

d) En el caso de una defensa de Tesis considerada Insuficiente (aunque haya sido aceptada para su exposición), el Tesista podrá solicitar por única vez una nueva fecha de exposición pasados 90 días de la defensa.

Artículo 15. Trámite de Expedición del Título y Registro de la Propiedad Intelectual

Para iniciar el trámite de expedición del título, el alumno deberá presentar dos ejemplares encuadernados de la versión final de su tesis (de acuerdo a lo establecido en el Art. 17, parte 2). En esta versión final deben incluirse las correcciones/modificaciones sugeridas por el Jurado.

La Facultad de Informática, a través de la Secretaría de Postgrado, realizará la inscripción del Registro de Propiedad Intelectual de la obra (tesis). El Registro tendrá como titular del mismo a la Facultad de Informática y como autor al Tesista (y sus directores).

Artículo 16. Formato de las Propuestas de Tesis de Maestría

a) Nombre y apellido del Alumno/Tesista. Maestría en la que está inscripto.
b) Nombre y apellido del Director/es y si correspondiera Codirector/es.
c) Título del Tema de Tesis propuesto.
d) Objetivo
   En este punto se indicará claramente y con una extensión no mayor a 400 palabras el objetivo general de la Tesis, los temas particulares que abordará y el aporte que resultará de su concreción.
e) Motivación /Estado del Arte del Tema
   En este punto se resumirá el contexto científico/tecnológico/académico que justifica el desarrollo de una Tesis en la temática. La extensión de este punto no debe exceder las 2 páginas. En ella se pueden hacer referencias/citas que refuercen la motivación que origina la propuesta.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

f) Temas de Investigación
Deben indicarse sintéticamente los temas centrales que el Tesista investigará en el desarrollo de la Tesis.

g) Desarrollos/Trabajo Experimental a Realizar
En el caso que la Tesis contemple la realización de trabajo experimental debe indicarse sintéticamente cual sería y el producto final (prototipo, mediciones, evaluaciones comparativas, etc) que resultará del trabajo propuesto, así como el impacto en el ámbito concreto de aplicación.

h) Esquema de Plan de Trabajo C/Actividades y Tiempos
Se indicarán las actividades principales del desarrollo de la Tesis y una distribución tentativa de tiempos. Debiera servir como un documento de control de la ejecución de la propuesta.

i) Posibilidades de Realización en el Ámbito del Tesista
Se puede indicar sintéticamente las posibilidades que tiene el Tesista en su contexto laboral (académico, profesional) para el desarrollo de la Tesis y si la misma está inserta en un proyecto de I/D específico apoyado por un organismo académico/científico/privado. En este punto se puede señalar algún aspecto metodológico que se considere importante para el desarrollo de la Tesis. También deben explicitarse los recursos con los que cuente para poder llevar adelante el desarrollo de la Tesis (por ejemplo, equipamiento, acceso a bibliografía específica, datos para realizar un muestreo particular para la investigación, etc.)

j) Bibliografía Básica Relacionada
Se citará la bibliografía relacionada más significativa. No se trata de una enumeración extendida de bibliografía sobre el tema general de la Tesis, sino una selección de textos/artículos/sitios WEB de referencia en el tema.

Artículo 17. Formato de las Tesis de Maestría

Parte 1

a) Las Tesis de Maestría en Ingeniería de Software deberán estar impresas en papel tamaño A4. La encuadernación tendrá tapa transparente y estará espiralada.

b) En la primera hoja del trabajo debe figurar:
Título del Trabajo de Tesis
Nombre y Apellido del tesista
Nombre y Apellido de Director/es y Codirector/es (si correspondiera)
"Tesis presentada para obtener el grado de Magister en Ingeniería de Software"
"Facultad de Informática - Universidad Nacional de La Plata"
Mes y año

c) La tesis deberá incluir un índice, un capítulo introductorio, un capítulo de conclusiones, nomenclatura y bibliografía única para todo el trabajo.

Parte 2

Con el objetivo de sistematizar la documentación de las Tesis de Maestría, y favorecer la difusión de las mismas y su consulta por alumnos de nuestra Facultad y de otras Unidades Académicas:

a) Las versiones definitivas de las Tesis de Maestría se imprimirán en un formato tipo libro con tapas duras al menos 2 ejemplares (1 para la biblioteca de la Facultad, otro para la Secretaría de Postgrado). Estas copias, que deberán ser presentadas una vez aprobada la Tesis, estarán a cargo del alumno.

b) En todos los casos deberá entregarse dos copias en soporte digital con la versión definitiva de la Tesis, de modo de poder poner el título y resumen en la página WEB de Postgrado y de la Facultad, y poder compartir el trabajo a pedido de los interesados en la información de la Tesis.

c) Si el autor de la Tesis está de acuerdo, se lo inscribirá en el Registro de la Propiedad intelectual a su nombre, con indicación de la realización en el ámbito de nuestra Facultad. El trámite estará a cargo de la Secretaría de Postgrado. Al mismo tiempo, el trabajo será publicado en la colección de trabajos de la Facultad dentro del repositorio del SEDICI.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Diseño de bases de datos

Duración: 108 hs. Totales.

Cantidad de horas presenciales/VC: 30hs
Cantidad de horas de actividades en línea y de trabajo final: 78hs.

- **OBJETIVOS GENERALES**

Discutir desde el punto de vista de la arquitectura de software, la problemática relacionada con el diseño y la implementación de sistemas orientados a objetos con persistencia en diferentes tipos de base de datos.

Analizar las nuevas herramientas, tecnologías y paradigmas para almacenar y recuperar eficientemente la información generada.

- **COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA**

C.1- Manejar y aplicar tecnologías actuales para el desarrollo de sistemas de software, incluyendo métodos, lenguajes, arquitecturas, frameworks y herramientas.

C.5- Diseñar y gestionar el almacenamiento de datos, aplicando diferentes tecnologías y frameworks de persistencia de datos actuales.

C.6- Tener capacidad de analizar el estado del arte en los distintos aspectos de la ingeniería de software, así como producir conocimiento científico en el área.

- **CONTENIDOS MINIMOS:**

- Persistencia Orientada a Objetos
- Alternativas de implementación
- Patrones de diseño utilizados en Persistencia
- Frameworks y productos

- **PROGRAMA**

-Persistencia Orientada a Objetos

- Principios de transparencia de la persistencia.
  - Requisitos.
  - Beneficios a nivel diseño y a nivel implementación.
  - Ortogonalidad de persistencia
- Persistencia por alcance
  - Concepto.
  - Ventajas y consecuencias de su aplicación.
  - Implementación.
- Transacciones
  - Manejo de transacciones bajo el paradigma OO.
  - Esquemas pesimista y optimista.
  - Demarcación de transacciones.
  - Transacciones explícitas vs implícitas.
- Versionamiento de instancias para control de concurrencia.
- Operaciones CRUD
  - Diferencias entre esquema tradicional y a partir de persistencia por alcance.
  - Optimización de performance

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

- Alternativas de implementación

- Mapeo Objeto-Relacional
  - Cualidades deseables de la integración
  - Performance sin compromiso
  - Transparencia.
  - Soporte para múltiples bases de datos.
  - Independencia del mapeador.
  - Estrategias de mapeo
    - Mapeo de los principales elementos del paradigma orientado a objetos
    - Mapeo de colaboradores
    - Mapeo de jerarquías
  - Optimización
  - Diferentes tipos de colecciones.
    - Mapeo de conjuntos y listas.
  - Lenguaje de consulta HQL
    - Introducción.
    - Principales elementos.
    - Path expressions.
    - Subconsultas.
    - Joins.
- Bases de datos NOSQL
  - Introducción
  - Comparación con las bases de datos relacionales
  - Diferentes tipos de bases de datos
    - Orientadas a documentos
    - Familias de columnas
    - Clave valor
    - Orientadas a grafos
  - Performance
  - MONGODB
    - Esquema de trabajo
    - Principales conceptos
      - Colecciones
      - Documentos
      - Réplicas
    - Consultas
    - Sharding

-Patrones de diseño utilizados en Persistencia

- Proxy: utilización del Proxy para optimización de performance. Lazy loading.
- Decorator: intercepción de mensajes para demarcación de transacciones.
- Repository: implementación de repositories para acceder eficientemente al modelo.
- Root Object: representación del sistema a través de clases dedicadas.
- DTO: transferencia de información entre las diferentes capas de la aplicación.
- DAO y razones para no utilizarlo: principios de diseño para capa de acceso a Datos. Consecuencias indeseadas de la aplicación de este patrón. Modelos anémicos.

-Frameworks y productos

- Spring framework.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

  - Inversión de control.
  - Inyección de dependencias
  - Configuración.
  - Springboot
- Hibernate
  - Configuración.
  - Ejemplos de uso.
- MONGODB
  - Instalación
  - Configuración
  - Pruebas.
- ELASTIC SEARCH
  - Instalación
  - Configuración
  - Pruebas

- **MODALIDAD DE EVALUACIÓN Y ACREDITACIÓN**

La materia se estructura en base a dos grandes ejes: teórico y práctico. Ambos ejes se llevan a cabo en su totalidad en modalidad virtual. A continuación, se introduce cada eje.

- **Eje teórico**: el cual involucra encuentros sincrónicos (a través del entorno virtual de enseñanza y aprendizaje -EVEA- de la Facultad) en los que se presentan y discuten los contenidos del Programa, incentivando la participación de los estudiantes a través del intercambio de experiencias y opiniones y fomentando así la construcción colectiva de conocimiento. Es requisito necesario contar con un porcentaje de 80% de asistencia a dichos encuentros sincrónicos, incluyendo el encuentro inicial de presentación de la materia, y el encuentro final de integración, ambos de asistencia obligatoria.
- **Eje práctico**: Este eje, implementa diferentes estrategias formativas, involucrando lectura de bibliografía, asignación de trabajos prácticos de carácter individual y otros de carácter grupal. El abordaje de este eje considera el uso de recursos como por ejemplo videos online, instalación de herramientas (software) y el desarrollo de un Proyecto integrador final.
  El seguimiento de este eje, se realiza de manera asincrónica a través del entorno virtual de enseñanza y aprendizaje (EVEA) de la Facultad. Eventualmente, se podrá usar como recurso el de videollamadas mediante alguna plataforma alternativa para este fin.

La evaluación se lleva a cabo a través de la aprobación de un Proyecto integrador. La propuesta de proyecto se presenta al finalizar el contenido del programa de la cursada y su planteo se da en relación a los temas abordados en el programa.
Dicho proyecto podrá realizarse en grupos de hasta 2 personas utilizando diferentes plataformas tecnológicas, las cuales serán acordadas con el docente.
El Proyecto se divide en cinco etapas. En la primera etapa, se debe presentar una línea base de requerimientos, la cual debe ser aprobada por el docente. Al tratarse de una propuesta del o de los estudiantes, favorece el aprendizaje significativo. La segunda etapa, consiste en presentar un diseño orientado a objetos, en el que al menos, se debe considerar un diseño de clases y de interacción para las principales funciones consideradas en la propuesta de la etapa 1. La tercera etapa, comprende la implementación de un subconjunto de los requerimientos aprobados en la etapa 1 y diseñados en la etapa 2. En esta etapa se despliegan estrategias de persistencia de la información en una base de datos a establecer de común acuerdo entre el grupo y el docente a cargo. En la cuarta etapa, se deberá redactar un informe donde se justifiquen, mediante un trabajo de relevamiento y análisis de bibliografía, los conceptos teóricos y las tecnologías empleadas durante el proyecto propuesto hasta la etapa anterior. Por último, la quinta etapa, considera un coloquio individual del proyecto elaborado.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Cada etapa del proyecto debe ser aprobada por el docente para avanzar a la etapa siguiente. En caso de desaprobar una etapa, se podrá realizar una reentrega que contemple las correcciones indicadas por el docente. El proyecto se considera aprobado, cuando sus cinco etapas han sido aprobadas.
El tiempo máximo para realizar el proyecto integrador desde el inicio del mismo, es de 6 meses.
En caso de desaprobar el proyecto integrador de la materia, el estudiante deberá recursarla.

- **RECURSOS Y MATERIALES DE ESTUDIO**

Se trabaja en base a materiales de estudio específicos tanto para el eje teórico como para el eje práctico.
El material de estudio es habilitado en el EVEA IDEAS con una frecuencia semanal:

- Textos digitales: textos digitales que recuperan aspectos teóricos.
- Videotutoriales: se han elaborado de manera específica para la materia videotutoriales orientados a mostrar procedimientos para la utilización de las herramientas que se abordan en el eje práctico.
- Videos de autores de referencia: se presentan algunos recortes de entrevistas a autores de referencia. Los mismos están disponibles en repositorios y se enlazan desde el área de Itinerario de IDEAS.
- Presentaciones multimedia y videos: desarrollados ad-hoc para abordar las diferentes temáticas de la materia.
- Guías de práctica: estas guías orientan la realización de actividades prácticas en las diferentes herramientas.

- **ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACION PLANIFICADAS PARA LA APROPIACIÓN DE LOS SABERES Y LA EVALUACIÓN**

Actividades prácticas:
Durante el dictado de la materia, los estudiantes llevarán adelante diferentes trabajos de experimentación que les permitirá poner en práctica los conceptos visitados durante las clases, así como emplear los frameworks y productos presentados, lo que les permitirá apropiarse de los contenidos.
Entre las principales actividades que deberán desarrollar los estudiantes se encuentran:

- Instalación de productos y experimentación: a lo largo de la materia se propone la instalación de diferentes productos para que los estudiantes tomen contacto con la tecnología y puedan desarrollar actividades de experimentación en base a consignas asignadas oportunamente.
- Acceso a videos y tutoriales: en algunas etapas del trabajo práctico se recomienda el acceso a materiales publicados en reconocidas plataformas para que el estudiante pueda formar criterio acerca de las diferentes alternativas a la hora de realizar el trabajo práctico.
- Proyecto práctico integrador: en forma individual o grupal se debe llevar adelante el desarrollo de un proyecto práctico en el que se pongan de manifiesto todos los conceptos vistos en la materia.
- Defensa del proyecto integrador: en un encuentro sincrónico final cada grupo/estudiante debe realizar una presentación y posterior defensa del proyecto realizado. Para la defensa, es necesario presentar tanto la funcionalidad implementada, así como también el código fuente desarrollado, También es necesario incluir diagramas que pudieran servir para presentar la arquitectura de software seleccionada. Esta defensa se realiza de manera virtual.

Investigación:
Los estudiantes deberán analizar artículos de investigación de la bibliografía complementaria y materiales que se presentan en forma semanal y estarán disponibles en IDEAS, con el fin de desarrollar un enfoque crítico de los distintos temas y complementar los conocimientos. Estos

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

artículos serán discutidos luego durante los encuentros sincrónicos y a través del espacio de foro de consultas de la plataforma virtual de enseñanza.

- **BIBLIOGRAFÍA BÁSICA**

- Atzeni, P., Bugiotti, F., Cabibbo, L., & Torlone, R. (2020). Data modeling in the NoSQL world. *Computer Standards & Interfaces*, *67*, 103149.
- Fisher, P., & Murphy, B. D. (2016). Spring Persistence with Hibernate. Apress.
- Juneau, J. (2020). Java Persistence Query Language. In Jakarta EE Recipes (pp. 527-567). Apress, Berkeley, CA.
- Keith, M., Schincariol, M., & Nardone, M. (2018). Pro JPA 2 in Java EE 8: An In-Depth Guide to Java Persistence APIs. Apress.
- Malhotra, R. (2019). Rapid Java Persistence and Microservices: Persistence Made Easy Using Java EE8, JPA and Spring. Apress.
- Meier, A., & Kaufmann, M. (2019). Nosql databases. In *SQL & NoSQL Databases* (pp. 201-218). Springer Vieweg, Wiesbaden.
- Milhalcea, V. (2016) High-Performance Java Persistence octubre de 2016 -ISBN-10: 973022823X
- Raj, P., & Deka, G. C. (2018). A Deep Dive into NoSQL Databases: The Use Cases and Applications. Academic Press.

- **BIBLIOGRAFÍA COMPLEMENTARIA**

- Acharya, B., Jena, A. K., Chatterjee, J. M., Kumar, R., & Le, D. N. (2019). NoSQL Database Classification: New Era of Databases for Big Data. International Journal of Knowledge-Based Organizations (IJKBO), 9(1), 50-65.
- Aniche, M., Yoder, J., & Kon, F. (2019, May). Current challenges in practical object-oriented software design. In 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER) (pp. 113-116). IEEE.
- Ansari, M.H., Tabatab Vakili, V. & Bahrak, B. Evaluation of big data frameworks for analysis of smart grids. *J Big Data* 6**,** 109 (2019). https://doi.org/10.1186/s40537-019-0270-8
- Cutler, J., & Dickenson, M. (2020). NoSQL Databases. In Computational Frameworks for Political and Social Research with Python (pp. 117-126). Springer, Cham.
- Dasadia, C. (2017). MongoDB Administrator's Guide: Over 100 practical recipes to efficiently maintain and administer your MongoDB solution. Packt Publishing Ltd.
- Fisher, P., & Murphy, B. D. (2016). Architecting Your Application with Spring, Hibernate, and Patterns. In Spring Persistence with Hibernate (pp. 1-16). Apress, Berkeley, CA.
- Mittal, M., Balas, V. E., & Hemanth, D. J. (Eds.). (2018). Data Intensive Computing Applications for Big Data (Vol. 29). IOS Press.
- Parmar, R. R., & Roy, S. (2018). MongoDB as an Efficient Graph Database: An Application of Document Oriented NOSQL Database. Data Intensive Computing Applications for Big Data, 29, 331.
- Rajput, D. (2017). Spring 5 Design Patterns: Master efficient application development with patterns such as proxy, singleton, the template method, and more. Packt Publishing Ltd.
- Reniers, V., Van Landuyt, D., Rafique, A., & Joosen, W. (2019). Object to NoSQL Database Mappers (ONDM): A systematic survey and comparison of frameworks. Information Systems.
- Shin, K., Hwang, C., & Jung, H. (2017). NoSQL database design using UML conceptual data model based on Peter Chen's framework. International Journal of Applied Engineering Research, 12(5), 632-636.
- Singh, K., Ianculescu, A., & Torje, L. P. (2018). Design Patterns and Best Practices in Java: A comprehensive guide to building smart and reusable code in Java. Packt Publishing Ltd.

Tópicos de Ingeniería de Software I

Duración: 108 hs. Totales.

Cantidad de horas presenciales/VC: 50hs.

Cantidad de horas de actividades en línea y de trabajo final: 58hs.

- **OBJETIVOS GENERALES**

Identificar, analizar y reconocer los aspectos principales de la Ingeniería de Requerimientos de Software.

Establecer el papel específico de los requerimientos en el proceso de desarrollo de los principales modelos de ciclo de vida del software

Conocer y analizar las principales técnicas utilizadas, evaluar su campo de aplicación y seleccionar las adecuadas.

Analizar el papel de los requerimientos en el proceso de desarrollo de software y en la mejora de dicho proceso.

Reconocer el papel de los requerimientos de software en los nuevos enfoques de desarrollo y en la utilización de nuevas tecnologías.

- **COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA**

C.1- Manejar y aplicar tecnologías actuales para el desarrollo de sistemas de software, incluyendo métodos, lenguajes, arquitecturas, frameworks y herramientas.

C.2- Tener capacidad para analizar diferentes modelos de proceso de desarrollo de software y evaluar su calidad tanto en aspectos del producto resultante como en la gestión de los individuos involucrados y sus interacciones.

C.3- Gestionar, planificar y controlar proyectos de software de distinta envergadura.

C.6- Tener capacidad de analizar el estado del arte en los distintos aspectos de la ingeniería de software, así como producir conocimiento científico en el área.

- **CONTENIDOS MINIMOS**

- Conceptos principales de la Ingeniería de Requerimientos del Software.
- Papel de los requerimientos en los ciclos de vida de desarrollo del software.
- Tipos de requerimientos
- Técnicas utilizadas en el proceso de Ingeniería de Requerimientos
- Nuevos enfoques y áreas de aplicación.

- **PROGRAMA**

Introducción a la Ingeniería de Requerimientos
Las dificultades esenciales del software. Problemática de la representación / modelo / descripción. Tipos de modelos. Dificultades para comprender los requerimientos. El gap semántico. El punto de vista de los diferentes stakeholders. Papel de los requerimientos en el proceso de desarrollo y en la calidad del software. Impacto positivo de las buenas prácticas de requerimientos. Impacto negativo de las fallas en requerimientos.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Requerimientos e Ingeniería de Requerimientos
Concepto de requerimientos. Clasificación. Necesidades, deseos y expectativas. Modelo de referencia de los requerimientos. Análisis de requerimientos. Requerimientos de la Empresa. Requerimientos funcionales. Requerimientos no funcionales. Requerimientos sobre datos. Requerimientos del usuario. Requerimientos del Sistema. Concepto de Ingeniería de Requerimientos. Alcance de la Ingeniería de Requerimientos.

Procesos de la Ingeniería de Requerimientos
Modelos a producir a lo largo del proceso de Requerimientos. Modelo de procesos de Loucopoulos. Procesos de requerimientos según el SWEBOK. Descripción general de los procesos de Elicitación, Especificación y Validación y sus relaciones. Los requerimientos en diferentes modelos de procesos. Gestión de requerimientos.

Elicitación de Requerimientos
Ubicación de la elicitación en el proceso global de requerimientos. Concepto de elicitación. El problema del sesgo del observador y su impacto en el contexto. Técnicas de elicitación: taxonomías y características de las técnicas. La Ingeniería de Requerimientos como proceso social.

Léxico Extendido del Lenguaje
Glosarios en la especificación de requerimientos. Léxico Extendido del Lenguaje. Objetivo y estructura. Categorización de símbolos. Descripción de las distintas categorías. Proceso de construcción. Buenas prácticas en la construcción del LEL. Descripción de noción e impactos.

Especificación de Requerimientos
Concepto de modelo y modelización. Concepto de Especificación de Requerimientos de Software. Motivación para su construcción. Usos y usuarios de la especificación. Estándar IEEE 830. Criterios que debe cumplir una Especificación de Requerimientos de Software.
Modelización de la empresa. Modelización mediante objetivos. Modelización de requerimientos funcionales. Modelización de requerimientos no funcionales.
Esquemas posibles de una especificación de requerimientos. Estándares de formato.

Validación de Requerimientos
Concepto de validación de requerimientos. Enfoques de sistemas expertos. Consistencia interna. Guías para la validación. Recursos necesarios para la validación. Técnicas disponibles: Prototipos, Simulación, Animación, Paráfrasis del lenguaje natural. Inspecciones de documentos de requerimientos: Proceso de inspección, Costos y beneficios

Gestión de Requerimientos
Concepto de Gestión de Requerimientos. Actividades y mejores prácticas de la Gestión de Requerimientos. Guías para definir requerimientos para la Gestión de Requerimientos. Políticas y prácticas para la trazabilidad, manual de trazabilidad. Gestión de cambio. Gestión de requerimientos volátiles y rechazados. Priorización de requerimientos. CMMI e Ingeniería de Requerimientos.

Requerimientos agiles
User Stories y otros productos agiles. Elicitacion y descubrimiento de requerimientos. Product Owner como miembro del equipo. Especificación y refinamiento. Priorización y gestión del backlog. Validación. Estimación y planning poker.

Temas avanzados de Ingeniería de Requerimientos
Negociación de requerimientos. Visualización de requerimientos. Reutilización de requerimientos. Tendencias en Ingeniería de Requerimientos. Requerimientos en aplicaciones web, móviles, ubicuas, context aware y Smart Cities. Ingeniería de Requerimientos orientada al

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

mercado. Software product lines.

- **MODALIDAD DE EVALUACIÓN Y ACREDITACIÓN**

La materia se desarrolla completamente en modalidad virtual a través de encuentros sincrónicos con actividades mediante el entorno virtual de enseñanza y aprendizaje (EVEA). Se requiere un 80% de asistencia a dichos encuentros, incluyendo el encuentro inicial de presentación de la materia, y el encuentro final de integración, ambos de asistencia obligatoria.

En esta materia se combinan las clases sincrónicas con actividades realizadas a través del entorno virtual de enseñanza y aprendizaje que propone el SIED.

Las clases son teórico – prácticas, en ellas se presentan los ejes temáticos, y luego se incluyen ejemplos prácticos, para evaluar/ejercitar y validar los conceptos presentados.

La evaluación de la materia es a través de trabajos prácticos parciales.

Los trabajos prácticos parciales son de alcance limitado, con el propósito de que los estudiantes investiguen, ejemplifiquen y desarrollen en mayor profundidad temas abordados en clase. Estos son realizados y aprobados durante la cursada de la materia.

Aprobados todos los trabajos prácticos, el estudiante debe rendir un examen final (escrito u oral) que se focaliza en los temas teóricos de la materia. La calificación obtenida resultará en la calificación final del curso.

- **RECURSOS Y MATERIALES DE ESTUDIO**

Como materiales de estudio, se dispone de:

- Presentaciones multimedia desarrolladas ad-hoc para introducir cada uno de los diferentes ejes temáticos.
- Ejemplos donde se aplican los conceptos teóricos
- Ejercicios prácticos que son desarrollados en clase
- Píldoras formativas con la explicación de algunos temas
- Material de lectura para investigar, profundizar y estudiar conceptos abordados en las clases
- Enlaces a artículos de actualidad de repositorios reconocidos en el área
- Libros digitales

También se presentan herramientas de software, utilizadas para mostrar y/o ejemplificar conceptos desarrollados en las clases sincrónicas.

- **ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACION PLANIFICADAS PARA LA APROPIACIÓN DE LOS SABERES Y LA EVALUACIÓN**

Actividades prácticas:

Desarrollo de trabajos prácticos parciales luego de cada eje temático de la materia. Estos trabajos serán ejercicios que comenzarán en clase y podrían finalizar en la misma clase o la siguiente. Estos trabajos tendrán una consigna que el docente explicará y luego, a partir de los conceptos previamente vistos, los alumnos tendrán que llevarlo a la práctica. Los trabajos podrán ser individuales o grupales. Para esto último se configurará el entorno virtual para que los alumnos del mismo grupo se encuentren en un espacio virtual diferente del resto. Durante el desarrollo del trabajo, el docente estará conectado respondiendo dudas y consultas.

Estos trabajos pretenden desarrollar y/o fortalecer las aptitudes de opinión crítica en los temas relativos del curso. Los alumnos deberán sintetizar una opinión como conclusión de cada trabajo. Los ejercicios grupales permiten que la opinión sea discutida entre los participantes del grupo y

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

así poder tener mejores argumentos.

También se pretende desarrollar la capacidad de poder comunicar y transmitir los resultados, en presentaciones pautadas a lo largo de la materia. Finalizada la actividad, se realizará una sesión de discusión conjunta donde los participantes comunicarán sus opiniones e intercambiarán los distintos puntos de vista.

Investigación:

Los alumnos buscarán, evaluarán y desarrollarán análisis crítico de libros y/o artículos de congresos científicos en temas relacionados a los dictados en la materia. De esta forma, se complementan los conocimientos con el enfoque de otros autores. El material lo podrían brindar los docentes, como así también, podría ser responsabilidad de los alumnos buscar material y decidir sobre la pertinencia y calidad del mismo.

- **BIBLIOGRAFÍA BASICA**

- Dick, J., Hull, E., Jackson, K., (2017) Requirements Engineering 4th edition, Springer.
- Koelsch, G., (2016) Requirements writing for system engineering, Apress.
- Adzic, G., (2011) Specification by example, Manning.
- Aurum, A., Wohlin, C,. (2005) Engineering and managing software requirements, Springer.
- Loucopoulos, P., Karakostas, V. (1995). System Requirements Engineering. Londres: McGraw-Hill.

- **BIBLIOGRAFÍA COMPLEMENTARIA**

El siguiente material lo conforman papers clásicos de la disciplina y otros más actuales reflejando las tendencias modernas

- Akbar M. A., Nasrullah, Shafiq, S., Ahmad, J., Mateen, M., and Riaz, M. T., "AZ-Model of software requirements change management in global software development," 2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), Quetta, pp. 1-6, 2018.
- Anuar, U., Ahmad, S., Emran, N. A., A simplified systematic literature review: Improving Software Requirements Specification quality with boilerplates, 2015 9th Malaysian Software Engineering Conference (MySEC), Kuala Lumpur, pp. 99-105, 2015.
- Pan, C., Lu, M., Zhang, H., Xu, B., "Qualitative Software Reliability Requirements: Concept, Classification and Practical Elicitation Methods," 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, pp. 164-171, 2018.
- Carrizo, D., Comparison of Research and Practice Regarding What We Mean by "The Right Software Requirements Elicitation Technique, 2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC), Lisbon, pp. 79-82, 2016.
- Chawla, S., Srivastava, S., A Goal based methodology for Web specific Requirements Engineering, 2012 World Congress on Information and Communication Technologies, Trivandrum, pp. 173-178, 2012.
- Creighton, O., Callele, D., Gotel, O., A summary of the fourth international workshop on Multimedia and Enjoyable Requirements Engineering (MERE'11), 2011 Fourth International Workshop on Multimedia and Enjoyable Requirements Engineering (MERE'11), Trento, pp. 4-8, 2011.
- Gotel, O., Morris, S., Requirements Tracery, in IEEE Software, vol. 28, no. 5, pp. 92-94, Sept.-Oct. 2011.
- Bjarnason, E., Borg, M., "Aligning Requirements and Testing: Working Together toward the Same Goal," in IEEE Software, vol. 34, no. 1, pp. 20-23, Jan.-Feb. 2017.
- IEEE, IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998 (Revision of IEEE Std 830-1993)
- ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering," in ISO/IEC/IEEE 29148:2018(E), pp.1-104, 30 Nov. 2018
- Jaramillo Franco, A., Requirements elicitation approaches: A systematic review, 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS), Athens, pp. 520-521, 2015.
- Leite, J., “Extreme Requirements (XR)”, Keynote at the Jornadas de Ingeniería de Requisitos Aplicadas, Sevilha, June, 2001,

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

- Lim, S. L., Finkelstein, A., StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation, in IEEE Transactions on Software Engineering, vol. 38, no. 3, pp. 707-735, May-June 2012.
- Mahaux, M., Nguyen, L., Gotel, O., Mich, L., Mavin, A., Schmid, K., Collaborative creativity in requirements engineering: Analysis and practical advice, IEEE 7th International Conference on Research Challenges in Information Science (RCIS), Paris, pp. 1-10, 2013.
- NASA, Instructional Handbook for Formal Inspections, https://sw-eng.larc.nasa.gov/files/2013/05/Instructional-Handbook-for-Formal-Inspections.pdf.
- Zhi, Q., Zhou, Z., Morisaki, S., Yamamoto, S., "An Approach for Requirements Elicitation using Goal, Question, and Answer," 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), Toyama, Japan, pp. 847-852, 2019.
- Schmid, K., Challenges and Solutions in Global Requirements Engineering -- A Literature Survey, in Software Quality. Model-Based Approaches for Advanced Software and Systems Engineering, Springer International Publishing, pp 85-99, 2014
- Spoletini, P., Ferrari, A., Requirements Elicitation: A Look at the Future Through the Lenses of the Past, 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, pp. 476-477, 2017.
- Thitisathienkul, P., Prompoon, N., Quality Assessment Method for Software Requirements Specifications Based on Document Characteristics and Its Structure, 2015 Second International Conference on Trustworthy Systems and Their Applications, Hualien, pp. 51-60, 2015.
- Wen, B., Luo, Z., Liang, P., Distributed and Collaborative Requirements Elicitation Based on Social Intelligence, 2012 Ninth Web Information Systems and Applications Conference, Haikou, pp. 127-130, 2012.
- Yozgyur, K., A proposal for a requirements elicitation tool to increase stakeholder involvement, 2014 IEEE 5th International Conference on Software Engineering and Service Science, Beijing, pp. 145-148, 2014.
- Oster, Z. J., "Improving Requirements Elicitation Through Listening Research," 2018 1st International Workshop on Learning from other Disciplines for Requirements Engineering (D4RE), Banff, AB, pp. 26-28, 2018.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Técnicas y Herramientas

Duración: 108 hs. Totales.

Cantidad de horas presenciales/VC: 30hs
Cantidad de horas de actividades en línea y de trabajo final: 78hs.

- **OBJETIVOS GENERALES:**

El objetivo de la materia es introducir a los alumnos en técnicas modernas de la programación, en particular en aspectos de la programación orientada a objetos, diseño orientado a objetos, técnicas complementarias como test de unidad, refactorings e introducción a patrones de diseño.

Se introduce también al alumno en el uso de un lenguaje de modelado gráfico orientado a objetos (UML), que le permitirá construir diagramas especificando distintos aspectos de un sistema.

Se enfatiza en la construcción de arquitecturas de software modulares, extensibles y reusables, que son conceptos claves para aplicaciones de gran porte.

- **COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA**

C.1- Manejar y aplicar tecnologías actuales para el desarrollo de sistemas de software, incluyendo métodos, lenguajes, arquitecturas, frameworks y herramientas.

C.2- Tener capacidad para analizar diferentes modelos de proceso de desarrollo de software y evaluar su calidad tanto en aspectos del producto resultante como en la gestión de los individuos involucrados y sus interacciones.

C.4- Definir parámetros de calidad tanto interna como externa de un producto software, y establecer procesos de evaluación y mejora que atiendan la satisfacción de todos los involucrados (el cliente, los usuarios y su experiencia, y el equipo de desarrollo).

C.6- Tener capacidad de analizar el estado del arte en los distintos aspectos de la ingeniería de software, así como producir conocimiento científico en el área.

- **CONTENIDOS MINIMOS**

- Principios del diseño orientado a objetos y lenguajes de modelado y programación orientados a objetos.
- Métodos ágiles. Testing y refactoring como prácticas esenciales de los métodos ágiles.
- Patrones de diseño.
- Mejora de la calidad interna del software

- **PROGRAMA**

Programación Orientada a Objetos

- Principios del Diseño Orientado a Objetos. Reutilización de software.
- Definición de clases y responsabilidades.
- Herencia y polimorfismo. Binding dinámico.
- Relaciones de conocimiento y composición. Objetos contenedores.
- Lenguajes orientados a objetos: v Maestriaariantes.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Lenguajes de modelado orientados a objetos
- El lenguaje de Modelado Unificado (Unified Modeling Language).
- Diagramas de Clases.
- Diagramas Dinámicos ó de Comportamiento: Diagramas de Interacción (Diagramas de Secuencia y Diagramas de Colaboración).

Testing de Unidad.
- Factores que favorecen la aparición de errores.
- Comparación entre diferentes alcances en el testing: Unidad, Funcional, Regresión.
- Frameworks para implementar Test de Unidad.
- Validación en tiempo de ejecución.

Patrones de diseño
- Introducción a patrones de diseño. Definición.
- Tipos de patrones de diseño de software.
- Catálogo “GoF” de patrones de diseño.

Refactoring.
- Definición.
- Análisis de los refactorings más importantes.
- Relación entre refactoring y patrones. Refactoring hacia patrones.

Métodos Ágiles.
- El módelo de cascada tradicional.
- Procesos de desarrollo basados en prototipos.
- Precepto fundamental sobre el cambio de código.
- Test Driven Development
- Elementos fundamentales de los métodos Ágiles: Usuarios, Testing, Integración Continua y Refactoring.
- Extreme Programming y Scrum.
- Herramientas de trabajo colaborativas que soporten las metodologías.

Calidad interna del software.
- Análisis de problemas de diseño y su solución mediante refactorings.
- Deuda técnica: costeo de rectificar los problemas

Aumentación Web.
- Introducción a la adaptación en aplicaciones Web
- Adaptación externa mediante aumentación Web
- Usos internos de adaptación Web

- **MODALIDAD DE EVALUACIÓN Y ACREDITACIÓN**

La evaluación de la materia es a través de trabajos prácticos parciales y un trabajo práctico final integrador. La calificación final del alumno es en base al promedio de los mismos. Los trabajos prácticos parciales son de pequeña envergadura y tienen una función poner rápidamente en práctica los conocimientos que se van abordando. Estos son realizados durante la cursada de la materia. El trabajo final integrador involucra la totalidad de técnicas y herramientas vistas durante el curso de la materia, y tiene un plazo máximo de entrega de 3 meses luego de la última clase. Los trabajos prácticos se realizan usando el lenguaje de modelado UML y diferentes lenguajes de programación para su implementación, tales como JavaScript, Smalltalk, etc. qué son los más apropiados de acuerdo a estos objetivos de aprendizaje/actualización.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Opcionalmente al trabajo final integrador, los alumnos podrán optar por un trabajo que requiera investigación en alguna de las temáticas abordadas durante el curso, sumado a una propuesta de aplicación de la misma.

Los trabajos prácticos pueden opcionalmente desarrollarse en grupos de 2 alumnos.

- **RECURSOS Y MATERIALES DE ESTUDIO**

El curso propone 15 encuentros sincrónicos en total. De este total, 13 de ellos son por videoconferencia, mediante el uso de la plataforma virtual de enseñanza Big Blue Button que permite diferentes maneras de interacción inmediata con los alumnos. Se requiere 80% de asistencia a los encuentros por VC (10 encuentros). Los 2 encuentros restantes tienen carácter obligatorio y presencial.

Los materiales de estudio son:

- Textos digitales: textos de lectura de referencia en en las temáticas tratadas durante el curso. Los textos de referencia son recuperados de la biblioteca del postgrado, revistas y/o de repositorios.
- Material preparado por los docentes del curso.
- Presentaciones digitales y materiales multimediales sobre el tema (de producción propia)

- **ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACIÓN PLANIFICADAS PARA LA APROPIACIÓN DE LOS SABERES Y LA EVALUACIÓN**

Actividades prácticas:

Desarrollo de trabajos prácticos parciales luego de cada eje temático de la materia. Estos trabajos están pensados para ser desarrollados en clases de taller, apuntando a una puesta en común.

Actividad 1: luego del dictado de los primeros temas, se trabaja sobre un modelo para que los alumnos puedan realizar un diseño e implementarlo sobre la tecnología a utilizar en el curso (JavaScript) para empezar a familiarizarse con la misma. Además, se trabajará con diagramas UML.

Actividad 2: luego de presentarse los contenidos sobre tests de unidad y la tecnología para programar y correr dichos tests en JavaScript, se realizará una actividad en la que los alumnos realizarán tests sobre código existente previamente utilizado en clase y también algunas funcionalidades con la metodología Tests-First.

Trabajo Final: se propone un trabajo final que abarca todos los temas trabajados durante las actividades prácticas intermedias. Se trata de un trabajo integrador donde se aplicarán patrones de diseño, técnicas ágiles de refactoring hacia patrones y test de unidad, en el contexto de artefactos de aumentación Web. Se tendrá encuentra particularmente la calidad interna alcanzada en el artefacto desarrollado.

El plazo máximo de entrega del trabajo final es de 3 meses a partir de la finalización del cursado de la materia.

Investigación:

Los alumnos analizarán artículos de investigación de la bibliografía complementaria con el objetivo de estudiar y comparar métodos ágiles y técnicas de programación orientada a objetos, y herramientas actuales de desarrollo, testing y refactoring. Estos artículos serán discutidos luego durante los encuentros sincrónicos y a través del espacio de foro de consultas de la plataforma virtual de enseñanza.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

- **BIBLIOGRAFÍA BASICA**

- Fowler, Martin. Refactoring: Improving the Design of Existing Code. Addison-Wesley, 2018.
- Rubin, K. Essential Scrum: A practical guide to the most popular Agile process. Addison Wesley. 2012.
- Antani, V., & Stefanov, S. Object-Oriented JavaScript. Packt Publishing Ltd. 2017.
- Timothy Budd. “An Introduction to Object-Oriented Programming”. Addisson-Wesley. (2008)
- Rebecca Wirfs-Brock. Object Design: Roles, Responsibilities, and Collaborations, Addisson-Wesley 2003.
- Gamma, Helm, Johnson, Vlissides. Design Patterns. Elements of Reusable Objects Oriented Software. Addison-Wesley, Professional Computing Series. 1995.
- Joshua Kerievsky. Refactoring to Patterns. Addison Wesley, 2004.
- Meszaros, G. xUnit test patterns: Refactoring test code. Pearson Education. 2007

- **BIBLIOGRAFÍA COMPLEMENTARIA**

- Matt Weisfeld. The Object-Oriented Thought Process, Third Edition, Pearson Education, Addison Wesley. (2008), ISBN-13: 978-0-672-33016-2
- Rebecca Wirfs-Brock. Design for Test. IEEE Software 26(5): 92-93 (2009)
- Gregor Hohpe, Rebecca Wirfs-Brock, Joseph W. Yoder, Olaf Zimmermann: Twenty Years of Patterns' Impact. IEEE Software 30(6): 88 (2013)
- Rebecca Wirfs-Brock: Designing with an Agile Attitude. IEEE Software 26(2): 68-69 (2009)
- Rebecca Wirfs-Brock: Looking for Powerful Abstractions. IEEE Software 23(1): 13-15 (2006)
- Martin Fowler, Kendall Scott. “UML Distilled”. Addisson-Wesley. (2004)
- Kent Beck and Cynthia Andres. Extreme Programming Explained. Addison-Wesley, 2005.
- Crockford, D. Javascript: the good parts: the good parts. " O'Reilly Media, Inc.". 2008.
- Kent Beck. Simple Smalltalk Testing: With Patterns. http://www.xprogramming.com/testfram.htm
- Stephane Ducasse. SUnit Explained. http://www.iam.unibe.ch/~ducasse/
- Chen, R., & Miao, H. (2013, June). A selenium based approach to automatic test script generation for refactoring javascript code. In 2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS) (pp. 341-346). IEEE.
- Burchard, E. (2017). Refactoring JavaScript: Turning Bad Code Into Good Code. " O'Reilly Media, Inc.".
- Fard, A. M., & Mesbah, A. (2013, September). Jsnose: Detecting javascript code smells. In 2013 IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM) (pp. 116-125). IEEE.
- Silva, L. H., Valente, M. T., & Bergel, A. (2017, May). Refactoring legacy JavaScript code to use classes: The good, the bad and the ugly. In International Conference on Software Reuse (pp. 155-171). Springer, Cham.
- Alves, N. S., Mendes, T. S., de Mendonça, M. G., Spínola, R. O., Shull, F., & Seaman, C. “Identification and management of technical debt: A systematic mapping study”. Information and Software Technology 70, 100-121. 2016.
- Kruchten, P., Nord, R. L., & Ozkaya, I. “Technical debt: From metaphor to theory and practice”. Ieee Software 29(6), 18-21. 2012.
- Letouzey, JL. “The SQALE Method for Evaluating Technical Debt”. In Third International Workshop on Managing Technical Debt (MTD) (pp. 31-36). IEEE. 2012
- Li, Z., Avgeriou, P., & Liang, P. “A systematic mapping study on technical debt and its management”. Journal of Systems and Software, 101, 193-220. 2015.
- Lim, E.; Taksande, N. & Seaman, C. “A Balancing Act : What Software Practitioners Have to Say about Technical Debt”, IEEE Software 29 (6), 22-27. 2012.
- Díaz, O., & Arellano, C. (2015). The augmented web: rationales, opportunities, and challenges on browser-side transcoding. ACM Transactions on the Web (TWEB), 9(2), 1-30.
- José Matías Rivero, Matias Urbieta, Sergio Firmenich, Mauricio Witkin, Ramón Serrano, Viviana Elizabeth Cajas, Gustavo Rossi:Improving Legacy Applications with Client-Side Augmentations. ICWE 2018: 162-176

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Tópicos de Ingeniería de Software II

Duración: 108 hs. Totales.

Cantidad de horas presenciales/VC: 40hs

Cantidad de horas de actividades en línea y de trabajo final: 68hs.

- **OBJETIVOS GENERALES**

Brindar a los asistentes técnicas y conceptos avanzados del desarrollo de Software, que permiten construir software robusto, mantenible, extensible y reutilizable. Se abordará el diseño de software desde una perspectiva arquitectural y de atributos de calidad, con énfasis en la reutilización de software a partir del concepto de frameworks. Se abordarán patrones de arquitectura, con un foco en el ámbito de aplicaciones Web. El diseño e implementación de aplicaciones interactivas será abordada a través de técnicas de maquetado y diseño de experiencia de usuario. Por otro lado se introducirán conceptos de inteligencia artificial y cómo puede aplicarse al campo de la ingeniería de software.

- **COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA**

C.1- Manejar y aplicar tecnologías actuales para el desarrollo de sistemas de software, incluyendo métodos, lenguajes, arquitecturas, frameworks y herramientas.

C.3- Gestionar, planificar y controlar proyectos de software de distinta envergadura.

C.4- Definir parámetros de calidad tanto interna como externa de un producto software, y establecer procesos de evaluación y mejora que atiendan la satisfacción de todos los involucrados (el cliente, los usuarios y su experiencia, y el equipo de desarrollo).

C.6- Tener capacidad de analizar el estado del arte en los distintos aspectos de la ingeniería de software, así como producir conocimiento científico en el área.

- **CONTENIDOS MÍNIMOS**

- Nociones de arquitectura de software y atributos de calidad
- Conceptos avanzados del diseño orientado a objetos y frameworks de desarrollo.
- Arquitecturas web y diseño de aplicaciones Web.
- Usabilidad, experiencia del usuario y estrategias de testing y mejora incremental
- Introducción a la Inteligencia Artificial, y su impacto sobre la Ingeniería de Software.

- **PROGRAMA**

Arquitectura y Atributos de Calidad

- Definición de arquitectura. Rol en el desarrollo de software.
- Atributos de calidad como conductores del diseño arquitectónico.
- Estilos arquitectónicos más comunes.
- Captura de decisiones arquitectónicas.
- Vistas de módulos, componentes y conectores, y despliegue.

Conceptos de frameworks

- Tipos de frameworks. Características.
- Principio de inversión de control. Hotspots.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

- Composición vs. Herencia en frameworks. Ejemplos.
- Relación con arquitecturas de referencia.

Arquitecturas Web

- Frameworks tradicionales multicapa. Model-View-Controller. Principios.
- Frameworks de aplicaciones Single-Page.
- Arquitecturas basadas en Microservicios.
- Nociones de infraestructura para puesta en producción.
- Contenedores y virtualización.

Usabilidad y UX en el proceso de desarrollo

- Relevamiento de requerimientos de interacción y experiencia de usuario (UX).
- Herramientas de maquetado y prototipado.
- Requerimientos en el contexto de metodologías ágiles.
- Patrones y buenas prácticas de diseño UX

Tests de usabilidad

- Pruebas de usuario remotas y análisis de resultados.
- Captura y reparación automatizadas de problemas de usabilidad web.
- Testing multivariado y A/B testing.

Agile + UX

- Técnicas de relevamiento y especificación orientadas a las tareas.
- Requerimientos de interacción temprano e iterativo en el ciclo de desarrollo ágil.
- Proceso de mejora incremental de la experiencia del usuario durante un desarrollo ágil, basado en las técnicas de refactoring y testing.

Introducción a la Inteligencia Artificial.

- Fundamentos. IA simbólica vs. IA no simbólica.
- IA no simbólica: Aprendizaje automático o machine learning. Aprendizaje supervisado vs. Aprendizaje no supervisado. Árboles de decisión. Deep Learning (Redes Neuronales Artificiales). Tipos de redes neuronales.
- Aplicaciones de la Inteligencia Artificial en el campo de la Ingeniería de Software.

- **MODALIDAD DE EVALUACION Y ACREDITACIÓN**

La evaluación de la materia es a través de trabajos prácticos parciales y de un trabajo práctico final integrador. La calificación final del alumno es en base al promedio de los mismos. Los trabajos prácticos parciales son de pequeña envergadura y tienen una función de poner rápidamente en práctica los conocimientos que se van abordando. Estos son realizados y aprobados durante la cursada de la materia.

El trabajo final integrador está basado en la elección por parte del alumno de alguna de las temáticas abordadas durante el curso y el estudio de la literatura actual; puede consistir en una extensión de alguno de los trabajos prácticos parciales y tiene un plazo máximo de entrega de 3 meses luego de la última clase. Los trabajos prácticos se realizan usando el lenguaje de modelado y diferentes lenguajes de programación para su implementación, tales como JavaScript, Smalltalk, Java, Python, etc. qué son los más apropiados de acuerdo con estos objetivos de aprendizaje/actualización.

- **RECURSOS Y MATERIALES DE ESTUDIO**

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

El curso propone 15 encuentros sincrónicos en total, a desarrollarse en su mayoría a través de videoconferencia, mediante el uso de la plataforma virtual de enseñanza Big Blue Button que permite diferentes maneras de interacción inmediata con los alumnos. Se requiere 80% de asistencia a los encuentros sincrónicos, incluyendo el encuentro inicial de presentación de la materia, y el encuentro final de integración, ambos de asistencia obligatoria.

Los materiales de estudio son:

- Textos digitales: textos de lectura de referencia en en las temáticas tratadas durante el curso. Los textos de referencia son recuperados de la biblioteca del postgrado, revistas y/o de repositorios.
- Material preparado por los docentes del curso.
- Presentaciones digitales y materiales multimediales sobre el tema (de producción propia)

- **ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACION PLANIFICADAS PARA LA APROPIACIÓN DE LOS SABERES Y LA EVALUACIÓN**

Actividades prácticas:

Desarrollo de trabajos prácticos parciales luego de cada eje temático de la materia. Estos trabajos están pensados para ser iniciados en clases prácticas estilo taller, apuntando a una puesta en común, que luego continuará individualmente cada alumno.

Actividad 1: luego del dictado de los primeros temas, se trabaja sobre un caso de estudio para que los alumnos puedan realizar un análisis de sus principales atributos de calidad y un (re-)diseño arquitectural del mismo, para luego implementarlo sobre alguna de las tecnologías a utilizar en el curso (por ej., Java). Además, se trabajará sobre la generación de diagramas de arquitectura basados en UML.

Actividad 2: al finalizar con los contenidos de UX + Usabilidad en el ciclo de desarrollo, se trabajará sobre un prototipo de aplicación desde el punto de vista del diseño de interacción. Los alumnos crearán un prototipo con las funcionalidades principales de la aplicación, empezando por el estudio de los potenciales usuarios, pasando por prototipos de baja fidelidad, ajustándolo en al menos dos iteraciones de pruebas de usuario.

Actividad 3: mediante esta actividad el alumno pondrá en práctica los conceptos teóricos aprendidos acerca de cómo diseñar e implementar un agente inteligente utilizando Redes Neuronales Artificiales RNAs a través de un framework.

El alumno logrará comprender el concepto de RNA con sus diferentes configuraciones de parámetros e hiperparámetros, incluyendo hiperparámetros relacionados con la estructura de la RNA e hiperparámetros relacionados con el entrenamiento. El alumno también será capaz de evaluar la calidad de la RNA para realizar su tarea.

Trabajo Final: el trabajo final consiste en la elección por parte del alumno de alguna de las temáticas abordadas durante las 3 actividades prácticas intermedias para ser ampliada. El desarrollo de este trabajo involucra el estudio de la literatura actual sumado a la extensión del trabajo de la actividad intermedia, con un grado medio de dificultad. Tiene un plazo máximo de entrega de 3 meses luego de terminada la cursada.

Investigación:

Los alumnos analizarán artículos de investigación de la bibliografía complementaria con el objetivo de profundizar en los temas abordados. Estos artículos serán discutidos luego durante los encuentros sincrónicos y a través del espacio de foro de consultas de la plataforma virtual de enseñanza. Eventualmente, los alumnos podrán plantear mejoras o enfoques alternativos para las problemáticas discutidas en los artículos.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

- **BIBLIOGRAFÍA BÁSICA**

- Cervantes, H., Kazman, R. Designing Software Architectures: A Practical Approach. Addison-Wesley. 2016
- Fayad, M. E., Schmidt, D. C., & Johnson, R. E. Implementing application frameworks: object-oriented frameworks at work. John Wiley & Sons, Inc..
- Fayad, M. E., Schmidt, D. C., & Johnson, R. E. (1999). Building application frameworks: object-oriented foundations of framework design. John Wiley & Sons, Inc.
- Antani, V., & Stefanov, S. Object-Oriented JavaScript. Packt Publishing Ltd. 2017.
- Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Addison Wesley, 1994.
- S. Russell y P. Norvig. Artificial Intelligence. A Modern Approach.Prentice Hall, 3ra edición. 2010.
- Nielsen, Jakob. Designing web usability: The practice of simplicity. New riders publishing, 1999.
- Krug, Steve. Don't make me think!: Web & Mobile Usability: Das intuitive Web. MITP-Verlags GmbH & Co. KG, 2018.

- **BIBLIOGRAFÍA COMPLEMENTARIA**

- Firmenich S., Bosetti G., Rossi G., Winckler M., Barbieri T. (2016) Abstracting and Structuring Web Contents for Supporting Personal Web Experiences. In: Bozzon A., Cudre-Maroux P., Pautasso C. (eds) Web Engineering. ICWE 2016. Lecture Notes in Computer Science, vol 9671. Springer,
- Grigera J., Garrido, A., Rivero, J.M., Rossi, G. (2017). Automatic detection of usability smells in web applications. Int. J. Hum. Comput. Stud. 97: 129-148 (2017)
- Firmenich S., Garrido A., Grigera J., Rivero M., Rossi G. Usability improvement through A/B testing and refactoring. Software Quality Journal 27(1): 203-240 (2019)
- Bouvin, N. O. (2019, September). From NoteCards to Notebooks: There and Back Again. In Proceedings of the 30th ACM Conference on Hypertext and Social Media (pp. 19-28).
- Martin Fowler: Domain-Specific Languages. The Addison-Wesley signature series, Addison-Wesley 2011, ISBN 978-0-321-71294-3, pp. I-XXVIII, 1-597
- Wolfgang Ertel:Introduction to Artificial Intelligence.Springer. ISBN 978-3-319-58486-7
- Matias Urbieta, Nahime Torres, José Matías Rivero, Gustavo Rossi, Francisco José Domínguez Mayo: Improving Mockup-Based Requirement Specification with End-User Annotations. XP 2018: 19-34
- Angular 6 by Example: Get Up and Running with Angular by Building Modern Real-world Web Apps, 3rd Edition Kevin D. Hennessy
- Designing with Data: Improving the User Experience with A/B Testing. Rochelle King, Elizabeth Churchill, Caitlin Tan. O'Reilly Media.2017
- Krug, Steve. Rocket surgery made easy: The do-it-yourself guide to finding and fixing usability problems. New Riders, 2009.
- Goodfellow, Ian; Bengio, Yoshua y Courville, Aaron (2016). Deeplearning. MIT Press.
- Keras: the Python deep learning API https://keras.io/
- Shai Shalev-Shwartz y Shai Ben-David. (Ed.). (2014).Understanding Machine Learning: From Theory to Algorithms. New York. UnitedState: Cambridge University Press.
- Exploring architecture blueprints for prioritizing critical code anomalies: Experiences and tool support. E Guimaraes, S Vidal, A Garcia, JA Diaz Pace, C Marcos. Software: Practice and Experience, 48 (5), 1077-1106. Wiley. (2018)
- High-Level Design Stories in Architecture-Centric Agile Development. J. A Diaz-Pace, A. J. Bianchi. 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), 137-144, IEEE. (2019)
- Assisting requirements analysts to find latent concerns with REAssistant. A Rago, C Marcos, J. A. Diaz-Pace. Automated Software Engineering, 23 (2), 219-252. Springer. (2016)

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Administración de Proyectos

Duración: 108 hs. Totales.

Cantidad de horas presenciales/VC: 50hs.
Cantidad de horas de actividades en línea y de trabajo final: 58hs.

- **OBJETIVOS GENERALES**

Identificar, analizar y establecer el alcance de aplicación de los conceptos principales de la Administración de Proyectos de Software.

Conocer los fundamentos de la Ingeniería de Software aplicables en forma directa a la Gestión de Proyectos de Software.

Subrayar los aspectos de la Ingeniería de Software y la Gestión de Proyectos, relacionados con la medición de productos y procesos y los enfoques experimentales.

Identificar y analizar el contexto de aplicación de los procesos de calidad y mejora de los procesos de las organizaciones

- **COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA**

C.1- Manejar y aplicar tecnologías actuales para el desarrollo de sistemas de software, incluyendo métodos, lenguajes, arquitecturas, frameworks y herramientas.

C.2- Tener capacidad para analizar diferentes modelos de proceso de desarrollo de software y evaluar su calidad tanto en aspectos del producto resultante como en la gestión de los individuos involucrados y sus interacciones.

C.3- Gestionar, planificar y controlar proyectos de software de distinta envergadura.

C.4- Definir parámetros de calidad tanto interna como externa de un producto software, y establecer procesos de evaluación y mejora que atiendan la satisfacción de todos los involucrados (el cliente, los usuarios y su experiencia, y el equipo de desarrollo).

C.6- Tener capacidad de analizar el estado del arte en los distintos aspectos de la ingeniería de software, así como producir conocimiento científico en el área.

- **CONTENIDOS MINIMOS**

- Conceptos principales de la Ingeniería de Software.
- Conceptos principales de la Administración de Proyectos de Software.
- Características de enfoques de desarrollo clásicos y ágiles
- Medición, calidad y mejora del proceso.

- **PROGRAMA**

Software e Ingeniería de Software
Concepto de software. Naturaleza y cualidades del software. Objetivo de la producción de software. Productos de software. Ingeniería de Software (IS). Concepto y panorama de la Ingeniería de Software.

Introducción a los Proyectos de Software

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Experiencia en la Gestión de Proyectos de Software. Características de los proyectos de Ingeniería. Razones de los fracasos y los éxitos en los proyectos de software. Unidad conceptual y complejidad dinámica de los proyectos. Importancia de la comunicación en un proyecto. Ley de Brooks.

Gestión de los Proyectos de Software
Introducción al Cuerpo de Conocimiento de la Gestión de Proyectos (PMBOK) del Project Management Institute (PMI). La Oficina de Proyectos (Project Management Office-PMO). Concepto de Proyecto y de Gestión de Proyectos. Procesos de la Gestión de Proyectos, su interacción y evolución a lo largo del proyecto. Áreas de conocimiento del PMBOK. Componentes y restricciones de un proyecto. Los procesos de calidad, riesgos y compras. Actividades del Gerente de Proyecto. Herramientas de gestión.

Ciclos de vida
Definición Modelo, proceso, técnica y herramienta. Ciclos de vida. Ciclo de vida del producto. Ciclo de vida del proyecto. Ciclo de vida de la gestión del proyecto. Integración de los distintos ciclos de vida.

Plan de Gestión de un Proyecto
Gestión de integración de un proyecto en el PMBOK. Desarrollo del plan de proyecto. Fundamentación de un plan de proyecto. Componentes de un plan de proyecto. Tareas, hitos y procesos. Formato y contenido de un Plan de Gerenciamiento de un Proyecto de Software.

Gestión del Alcance y detalle de tareas
La Gestión del Alcance de un proyecto como área de conocimiento del PMBOK. Planeamiento y control del alcance en un proyecto. Alcance del producto, de los procesos y del proyecto. El documento de alcance de un proyecto. La estructura de desglose tareas del proyecto (WBS). Guías prácticas para la construcción. Procesos de aseguramiento de calidad del WBS. Definición de workpackages.

Gestión del Tiempo
La Gestión del Tiempo como área de conocimiento del PMBOK. Procesos de planeamiento y control del tiempo en un proyecto. Actividades en un paquete de trabajo. Contenido de las actividades de gestión del tiempo. Organización de las tareas de gestión de tiempo. Relación entre duración y esfuerzo. Datos para estimar el tiempo disponible. Recursos: definición y clasificaciones. Técnicas y herramientas para estimar tiempos y esfuerzos de recursos. Documentos para el planeamiento y control.

Estimaciones y tamaño del software
Incertidumbre de las estimaciones. Fuentes de errores en las estimaciones. Errores relacionados con la práctica de estimación. Manejo de los errores de estimación. Factores determinantes del esfuerzo, costo, cronograma. Tamaño del software. Medidas de proceso, producto y recursos. Medición de atributos externos e internos del software. Medidas de longitud, funcionalidad, complejidad y reuso.

Control de proyectos
Características generales de los controles. Control Gerencial. Informes de avance. Control de proyectos. Earned Value Analysis (EVA). La medición del avance de un proyecto. Componentes principales del Earned Value Analysis. Variaciones y ratios. Implementación de EVA.

Calidad y Mejora del proceso software
Calidad del software. Procesos de calidad del PMI. Introducción al Modelo de Maduración de Capacidad. Objetivos del modelo clásico de Capability Maturity Model (CMM). Niveles de

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

madurez. Evolución del proceso. Estructura del CMM. Áreas clave del proceso por nivel. Capability Maturity Model Integration (CMMI). Otros modelos de calidad.

Gestión de riesgos.
Concepto de gestión de riesgo: amenazas y oportunidades. Definiciones: incertidumbre, probabilidad, impacto. Actividades: identificación, análisis cualitativo, análisis cuantitativo, planificar respuesta.

Capital humano
Recursos y recursos humanos. Modelos de producción industrial. Canales de comunicación. Comunicación no verbal. Sesgos en la comunicación. Desafíos de la interculturalidad. Equipos: tipología y desarrollo. Personalidades: tipología y motivación. Conducción: estilos, liderazgo y poder. Negociación.

Gestión de proyectos agiles
Metodologías ágiles. Manifiesto ágil. Principios y valores. Equipos auto-organizados. Liderazgo. XP. SCRUM. Kanban. Kaizen.

- **MODALIDAD DE EVALUACIÓN Y ACREDITACIÓN**

La materia se desarrolla completamente en modalidad virtual a través de encuentros sincrónicos con actividades mediante el entorno virtual de enseñanza y aprendizaje (EVEA). Se requiere un 80% de asistencia a dichos encuentros, incluyendo el encuentro inicial de presentación de la materia, y el encuentro final de integración, ambos de asistencia obligatoria.

La evaluación se realiza en dos etapas. Durante el desarrollo del curso se realizan actividades prácticas que constituyen evaluaciones parciales. Y al final del curso se realiza una evaluación final integradora.

Las actividades prácticas durante el curso tienen distintas mecánicas: trabajos individuales y trabajos grupales. Los trabajos individuales apuntan principalmente a poner en práctica algún método, técnica o herramienta. Mientras que los trabajos grupales (ya sean de 2 o más integrantes) tienen por objetivo desarrollar una opinión crítica sobre algún método, técnica o herramienta. Es allí en donde el discutir con un par es de mucha utilidad. La disciplina de gestión de proyectos es una disciplina cuya aplicación depende del contexto, por lo cual, el intercambio de opiniones es una parte muy valiosa del curso.

Aprobadas las actividades prácticas, los estudiantes deberán rendir un examen final integrador en el cual deberán mostrar conocimiento sobre los temas dictados, como así también su opinión y punto de vista.

- **RECURSOS Y MATERIALES DE ESTUDIO**

El curso dispone de los siguientes recursos y materiales:
- Slides powerpoint para presentar cada uno de los diferentes temas.
- Ejemplos de aplicación.
- Ejercicios prácticos para desarrollar en clase.
- Píldoras formativas para explicar temas puntuales que lo requieran.
- Material de lectura complementaria (libros, articulos cientificos o articulos de divulgación), tanto para profundizar los conceptos vistos en clase, como así tambien para realizar actividades prácticas.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

- **ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACION PLANIFICADAS PARA LA APROPIACIÓN DE LOS SABERES Y LA EVALUACIÓN**

Actividades prácticas

Los alumnos deben realizar Trabajos Prácticos (individuales y grupales) en cada eje temático. Estas actividades pueden tomar distintas formas: (i) poner en práctica cierto método, técnica o herramienta, (ii) sintetizar una opinión sobre cierto método, técnica o herramienta (iii) comunicar tanto las características del método, técnica o herramienta, como la opinión sintetizada.

(i) Las actividades prácticas de cierto método, técnica o herramienta serán ejercicios que comenzarán en clase y podrían finalizar en la misma clase o la siguiente. Estas actividades tendrán una consigna que el docente explicará y luego, a partir de los conceptos previamente vistos, los alumnos tendrán que llevarlo a la práctica. Las actividades podrán ser individuales o grupales. Para esto último se configurará el entorno virtual para que los alumnos del mismo grupo se encuentren en un espacio virtual diferente del resto. Durante el desarrollo de la actividad, el docente estará conectado respondiendo dudas y consultas.

(ii) Ya sea en forma explícita (a través un ítem dentro de las consignas) o en forma implícita, los alumnos deberán sintetizar una opinión como conclusión de cada actividad. Los ejercicios grupales permiten que la opinión sea discutida entre los participantes del grupo y así poder tener mejores argumentos.

(iii) Finalizada la actividad, se realizará una sesión de discusión conjunta donde los participantes comunicarán sus opiniones e intercambiarán los distintos puntos de vista.

Investigación:

Con el fin de enriquecer la actividad (ii) mencionada en el apartado anterior, los alumnos deberán recurrir al análisis e investigación de material complementario. Este análisis consiste en identificar y evaluar métodos, técnicas y herramientas similares a la indicada por la actividad (y vista en el eje temático), de forma de complementar los conocimientos con el enfoque de otros autores. El material complementario lo podrían brindar los docentes, como así también, podría ser responsabilidad de los alumnos buscar material y decidir sobre la pertinencia y calidad del mismo.

- **BIBLIOGRAFÍA BASICA**

Nicholas, John M., Steyn, Herman: Project Management for Engineering, Business and Technology, Taylor and Francis, 6th edition 2021

Ruhe, Günther, Wohlin, Claes: Software Project Management in a Changing World, Springer, ISBN 978-3-642-55034-8, © 2014.

Project Management Institute: A guide to the Project Management Body of Knowledge (PMBOK guide) 6th edition, ISBN 9781628251845, © 2018.

Robert K. Wysocki: Effective Project Management: traditional, agile, extreme 7th edition, John Wiley & Sons © 2014

Bourque, Pierre, Fairley, Richard: Guide to the software engineering body of knowledge, SWEBOK v3.0, IEEE, ISBN 0-7695-5166-1, © 2014.

Jason, Jennifer: Agile project management: kanban, scrum, kaizen, ISBN 9781719949187, 2018.

Boehm, B, Rombach, H. D, Zelkowitz, M. V. Foundations of Empirical Software Engineering, The Legacy of Victor R. Basili, Springer-Verlag Berlin Heidelberg, 978-3-540-24547-6, 2005.

Brooks, F., The Mythical Man-month. Essays on Software Engineering, 1982, Addison Wesley, Reading, Massachusetts.

- **BIBLIOGRAFÍA COMPLEMENTARIA**

El siguiente material lo conforman papers clásicos de la disciplina y otros más actuales reflejando las tendencias modernas

Bloch M, Blumberg S, Laartz J Delivering large-scale IT projects on time, on budget and on value. McKinsey Finance 45:28–35, 2013Dalcher D The nature of project management: a reflection on the anatomy of major projects by Morris and Hough. Int J Manag Proj Bus 5(4):643–660, 2012.

Delater A, Narayan N, Paech B, Tracing requirements and source code during software development. In: ICSEA'12: 7th international conference of software engineering advances, pp 274–282, 2012.

Dingsøyr T, Nerur S, Balijepally V, Moe NB, A decade of agile methodologies: towards explaining agile software development. J Syst Softw 85(6):1213–1221. doi:10.1016/j.jss.2012. 02.033, 2012.

Eveleens JL, Verhoef C, The rise and fall of the chaos report figures. IEEE Softw 27 (1):30–36, 2010.

Hayat, F., Rehman, A. U., Arif, K. S., Wahab, K. Abbas, M., "The Influence of Agile Methodology (Scrum) on Software Project Management," 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Toyama, Japan, pp. 145-149, 2019.

Gibbs, W.W., "Software Chronicle Crisis", Scientific American, November 1994

Gotel, O., Cleland-Huang, J., Huffman Hayes, J., Zisman, A., Egyed, A., Grunbacher, P., Dekhtyar, A., Antoniol, G., Maletic. J., The grand challenge of traceability (v1.0). In Software and Systems Traceability, pages 343–409. Springer, 2012.

ISO/IEC/IEEE International Standard - Systems and software engineering - Life cycle processes - Project management," in ISO/IEC/IEEE 16326:2019(E) , vol., no., pp.1-42, 13 Dec. 2019

Jalali S, Wohlin C, Global software engineering and agile practices: a systematic review. J Softw Evol Proces 24(6):643–659, 2012.

Jørgensen M, Moløkken K, How large are software cost overruns? A review of the 1994 Chaos report. Inform Softw Technol 48(8):297–301, 2006.

Kennedy D, Nur M, The rise of taylorism in knowledge management. In: Proceedings of PICMET'12: technology management for emerging technologies (PICMET), 2012.

Kruchten P, The frog and the octopus – a model of software development. CSI Commun 35 (4):12–15, 2011.

Lampasona C, Heidrich J, Basili V, Ocampo A, Software quality modeling experiences at an oil company. In: Proceedings of the 6th international conference on empirical software engineering and measurement (ESEM), 20–21, pp 243–246, 2012.

Larman C, Basili VR, Iterative and incremental development: a brief history. Computer 36 (6):47–56, 2003.

Moe NB, Dingsøyr T, Dyba° T, A teamwork model for understanding an agile team: a case study of a Scrum project. Info Softw Technol 52(5):480–491, 2010.

Moe NB, Smite D, Hanssen GK, From offshore outsourcing to offshore insourcing: three stories. In: Proceedings of the 7th international conference on Global Software Engineering (ICGSE), pp 1–10, 2012.

Saleem, N., "Empirical Analysis of Critical Success Factors for Project Management in Global Software Development," 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE), Montreal, QC, Canada, pp. 68-71, 2019.

Novais RL, Torres A, Mendes TS, Mendonc¸a M, Zazworka N, Software evolution visualization: a systematic mapping study. Inf Softw Technol 55:1860–1883, 2013

Noll J, Beecham S, Richardson I. Global software development and collaboration: barriers and solutions. ACM SIGCSE bulletin - special section on global intercultural collaboration, September 2010.

Steinberga L,Smite D (2011) Towards a contemporary understanding of motivation in distributed software projects: solution proposal, vol 770. University of Latvia, Computer Science and Information Technologies, pp 15–26, 2011.

Wong, W. Y., Wen Yu, S., Too, C. W., "A Systematic Approach to Software Quality Assurance: The Relationship of Project Activities within Project Life Cycle and System Development Life Cycle," 2018 IEEE Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia, pp. 123-128, 2018.

Metodología de la Investigación

Duración: 70 hs. Totales.

Cantidad de horas presenciales/VC: 25hs

Cantidad de horas de actividades en línea y de trabajo final: 45hs.

- **OBJETIVOS GENERALES**

Que los alumnos sean capaces de:

- Reconocer los fundamentos teóricos para desarrollar proyectos de investigación, o de empresa, aplicando la metodología científica de una forma rigurosa.

- Conocer la metodología científica, adquirir destreza en la búsqueda de información contrastada referente a un tema de investigación y ser capaz de comunicar con fluidez y rigurosidad los resultados de su trabajo.

- Conocer cuestiones tanto legales como éticas sobre sus acciones como investigador y adquirir competencias de trabajo en equipo.

- Adquirir habilidades sobre cómo diseñar un proyecto de investigación, en la búsqueda de fuentes de financiación, planificación, presentación de resultados y en la transferencia a la industria de los resultados de la investigación.

Luego de un tratamiento general, se desarrollan módulos específicos para el Doctorado y cada uno de los Magíster, coordinados por los Directores de cada carrera.

- **PROGRAMA**

Bloque 1: Conceptos fundamentales sobre la investigación científica y tecnológica.

Los fines de la investigación. Tipos de investigación: Según el objetivo. Según los métodos. Métodos y técnicas de investigación. Diseño de un proyecto de investigación.

Bloque 2: Las fuentes de información: Documentación científica.

Concepto de información contrastada: El proceso de revisión. Fuentes de información (Revistas, Congresos, Bases de datos, Internet).
Criterios de valoración de las fuentes de información: Los “citation indexs”. La escritura técnica. Organización de artículos e informes.
Comunicaciones en congresos, reuniones, etc. (presentaciones orales y póster).
Lectura crítica de artículos (El proceso de revisión).

Bloque 3: La financiación de la investigación.

Investigación pública y privada. Centros de investigación: Las Universidades. Los centros de investigación públicos, Ayudas a la investigación.
Programas de investigación: Política científica en la Argentina.
Programas de cooperación científica internacional (país-país, redes, Iberoamérica)
Becas y ayudas. Otros programas.
La transferencia de tecnología y su relación con la Investigación.

Bloque 4. El equipo de investigación

Organización y estructura.
Dependencia entre proyectos.
Interdisciplinariedad (equipos multi-centros).

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Bloque 5. La ética en la investigación

Grandes problemas éticos de la investigación científica en la actualidad.
Normas éticas (IEEE, ACM)
Investigación y sociedad.
El fraude científico.
La protección de los inventos. Patentes. Propiedad intelectual. Derechos de autor.

Bloque 6: La Tesis Doctoral y la Tesis de Magister

Objetivos. Aspectos formales de cada nivel de Tesis.
Análisis de los requerimientos en el caso de la Facultad de Informática de la UNLP.
Organización y estructura de una Tesis de Doctorado. Organización y estructura de una Tesis de Magister.
Bibliografía (Estilos de referencia).
El rol del Doctor en Ciencias Informáticas. El rol del Magister en Informática.

Bloque 7: Metodología de Investigación específica por carrera (dictado en forma separada por carrera).

Ámbito de aplicación de las Tesis de Magister en cada caso.
Validación del aporte y los resultados experimentales de la Tesis de Magister.
Análisis comparado de Tesis de Magister en el área.
Métodos de seguimiento y evaluación en el proceso de elaboración de la Tesis de Magister en cada carrera.
Puntos críticos en la redacción de la Tesis de Magister.
Defensa de proyectos de Tesis.

- **MODALIDAD DE EVALUACIÓN Y ACREDITACIÓN**

Se realiza una evaluación formativa a través de la participación en espacios de consulta del EVEA y de entregas que dan cuenta del avance en la planificación de un trabajo de investigación y desarrollo que deben realizar los alumnos a lo largo del proceso.

Se propone un encuentro inicial para la presentación del curso y una de defensa de los trabajos realizados al final de curso. Este último se trata de una evaluación sumativa, mediante la entrega y defensa en plenario de dicha propuesta de proyecto de investigación y desarrollo vinculada al área temática de la carrera.

Además, los alumnos deben participar de los encuentros propuestos en forma obligatoria, ya sea por VC y/o en forma presencial.

- **RECURSOS Y MATERIALES DE ESTUDIO**

El curso combina encuentros sincrónicos presenciales y/o por VC, con espacios de debate y seguimiento tutorial a través del EVEA IDEAS. Se realizan tutorías personalizadas de una actividad de diseño de propuesta de proyecto de investigación y desarrollo. Además, se aborda la lectura de artículos, proyectos de investigación como ejemplo, acceso a revistas para analizar el citation index, y consultas vía la mensajería de IDEAS.

Como materiales de estudio se presentan:

- Artículos científicos de revistas disponibles en los repositorios de ACM y de IEEE.
- Presentaciones multimedia desarrolladas por los docentes.
- Ejemplos de proyectos de investigación para analizar.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

- Ejemplos de propuestas de tesis de maestría para su lectura crítica
- **ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACIÓN PLANIFICADAS PARA LA APROPIACIÓN DE LOS SABERES Y LA EVALUACIÓN**

Las principales actividades planificadas son:

- Exposición dialogada durante los encuentros sincrónicos.
- Dinámicas de debate con el planteo de casos y ejemplos para su análisis.
- Análisis de índices de citación en revistas reconocidas del área.
- Diseño de una propuesta de proyecto de investigación y desarrollo, con entregas parciales a lo largo del proceso. Las entregas se realizan mediante la mensajería de IDEAS.
- Entrevistas con la directora de la carrera para consultas específicas

- **BIBLIOGRAFÍA**

Los docentes han desarrollado y recopilado material de estudio específico en formato digital para el desarrollo del curso. Estos materiales se actualizan previo al dictado del curso todos los años.

Además, se trabaja con tesis disponibles en el SEDICI, modelos de presentación de proyectos de investigación, artículos científicos de revistas internacionales disponibles en IEEE, ACM y Springer. También se analizan trabajos de la revista TEyET y de congresos.

- Avila Baray, H. L. (2006) Introducción a la metodología de la investigación Edición electrónica.http:// www.eumed.net/libros/2006c/203/
- Bunge, M. (2002). La investigación científica: su estrategia y su filosofía. 2a.ed. Buenos Aires: Siglo XXI Editores Argentina.
- Cataldi, Z.; Lage, F. J. (2004) Diseño y organización de tesis. Buenos Aires: Nueva Librería
- Cea D´Ancona Á (1997), Métodos y Técnicas de Investigación cuantitativa, Editorial Síntesis Madrid.
- Cohen, N. (Compilador); Piovani, J. I.(Compilador) (2008). La metodología de la investigación en debate. La Plata, BA: EDULP.
- De Bono, E. (2007) El Pensamiento Lateral. Manual de creatividad. Paidós
- Eco, U. (1998) Cómo se hace una tesis: técnicas y procedimientos de estudio, investigación y escritura. Barcelona: Gedisa, 267 páginas
- Hernández Sampieri, R., Fernández Collado, C. y Baptista Lucio, R. (2007). Metodología de la Investigación. McGraw Hill.
- Iñiguez, L. (2004): El debate sobre metodología cuantitativa versus cualitativa. Universidad Autónoma de Barcelona: http://antalia.uab.es/liniguez/
- Marradi, A.; Archeri, N. y Piovani, H. (2007)- Metodología de las Ciencias Sociales. -- Buenos Aires: Emecé,.
- Pérez Serrano, G. (2002) Investigación cualitativa. II Retos e interrogantes: Técnicas y análisis de datos. Editorial la Muralla.
- Sabino, C. (2006) El proceso de investigación, Lumen-Humanitas, Bs.As., 1996..
- Sautu, R. (Compilador). (2007) Práctica de la investigación cuantitativa y cualitativa: articulación entre la teoría, los métodos y las técnicas. Buenos Aires: Lumiere.
- Sierra Bravo, R. (2005). Tesis doctorales y trabajos de investigación científica: metodología general de su elaboración y documentación. 5a. ed. Madrid: Thomson Editores
- Tamayo y Tamayo, M. (2007). El proceso de la investigación científica: incluye evaluación y administración de proyectos de investigación. 4a.ed. México, DF: Limusa,.
- Vasilachis de G, I.; Ameigeiras, Aldo R.; Chernobilsky, Lilia B.; Gimenez Beliveau, V. (2006) Estrategias de investigación cualitativa. Buenos Aires.
- Yin, R. (1994) Case study research, Design and Methods. 2da. Ed. Sage Publications. Inc. Thousands Oaks.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Taller de Tesis

Duración: 50 hs. Totales.

Cantidad de horas presenciales/VC: 20hs
Cantidad de horas de actividades en línea y de trabajo final: 30hs.

- **OBJETIVOS**

***Objetivos generales**:*

- Diferenciar características generales de informes, papers, trabajos de especialización, tesis maestría y doctorado en términos de sus características formales y metodológicas.

- Presentar distintos tipos de diseño aplicables según el tipo de proyectos.

- Orientar para la elaboración de anteproyectos de informe o tesis que contemplen aspectos teóricos y metodológicos.

Objetivos específicos:

- Desarrollar las destrezas de búsqueda, evaluación y organización de la literatura científica.

- Profundizar el conocimiento de la metodología requerida por el trabajo de tesis

- Adquirir y dominar los estilos de redacción de trabajos científicos y de su presentación oral.

- Elaborar un posible del plan de tesis, orientado a la carrera específica del alumno.

Esta asignatura se vincula con los objetivos de la carrera al presentar una metodología y herramientas para la redacción de la Tesis de Maestría.

- **PROGRAMA**

Módulo I: El objeto de estudio de la investigación.

Panorama de la investigación en el área donde se va a trabajar. Análisis de la literatura publicada para identificar tendencias, avances, problemas sin resolver.
Actividades para la revisión de la literatura pertinente.
Construcción de la fundamentación. Justificación y alcances.
Las citas bibliográficas. El plan de tesis.

Módulo II: El marco teórico de la investigación.

Estado de la cuestión. Formulación de hipótesis.
Definición de objetivos: general y específicos.
Coherencia entre el problema, la fundamentación y los objetivos y la hipótesis.

Módulo III: El diseño metodológico.

Elección del diseño. Operacionalización de la investigación.
Tipos de diseño. Criterios de selección de un diseño de investigación.
Recolección de datos y tipos de análisis de los mismos.
Elección de la muestra.
Cronograma de trabajo.

Módulo IV: La presentación de los resultados.

Análisis e interpretación de los resultados.
Formulación de las conclusiones. La validación de los resultados.
El artículo para revistas científicas.
Redacción de tesis. El estilo científico. El proceso de escritura académica.

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

Módulo V: Las partes principales de la tesis.

Resumen. Introducción, Fundamentación
Objetivo/s, Metodología, Resultados. Conclusiones y logro de objetivos planteados. Futuras líneas de investigación. La legitimidad de las fuentes consultadas. Comunicación de los resultados de la investigación.

- **MODALIDAD DE EVALUACIÓN Y ACREDITACIÓN**

El alumno deberá participar de las actividades propuestas durante los encuentros sincrónicos y las que se realizan mediadas por el entorno virtual de enseñanza y aprendizaje: lectura de materiales de referencia, consultas, y desarrollo del trabajo que acompaña todo el taller en forma posterior al primer encuentro.

De acuerdo al grado de avance de cada alumno con su propio proyecto de tesis, se le ofrecen dos alternativas de trabajo para evaluar y acreditar el taller:

A- Presentación del informe técnico con el análisis de una propuesta dada por los docentes del taller, para encontrar puntos fuertes, débiles de la propuesta y realizar recomendaciones de mejora.

Defender este informe en el marco del último encuentro del taller.

B- En caso de tener un proyecto de tesis ya planificado y/o en progreso se sugiere completar una propuesta de plan de tesis con sus partes constitutivas y realizar una defensa de éste en el encuentro final del taller.

- **RECURSOS Y MATERIALES DE ESTUDIO**

El curso propone 2 encuentros presenciales: uno al inicio de presentación de la propuesta e indagación de las necesidades de cada alumno; y otro al final, para la defensa de los trabajos finales. En el proceso se realizan encuentros por VC y/o presenciales (según las necesidades de cada alumno) para realizar tutorías sobre la realización de las actividades, consultas a través de la mensajería del EVEA IDEAS, lectura de material de referencia, e indagación de proyectos de tesis en el repositorio SEDICI de la UNLP.

Como materiales de estudio se presentan:

- Artículos científicos.
- Material de estudio desarrollado por los docentes.
- Ejemplos de propuestas de tesis de maestría.
- Proyectos de tesis del repositorio SEDICI.

- **ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACIÓN PLANIFICADAS PARA LA APROPIACIÓN DE LOS SABERES Y LA EVALUACIÓN**

Las principales actividades planificadas son:

- Lectura crítica de artículos y materiales de estudio preparados por los docentes
- Consultas y tutoría vía la mensajería del EVEA según los avances de cada alumno
- Entrega del trabajo de cada alumno via la mensajería del EVEA.
- Devolución a distancia de los trabajos corregidos, con sugerencias de mejora.
- Presentación y defensa de los anteproyectos y los estudios analíticos en el encuentro presencial final.
- Entrevistas con la directora de la carrera para consultas específicas

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar

- **BIBLIOGRAFÍA**

- Carlino, P. (2008) El proceso de escritura académica. Cuatro dificultades en la enseñanza universitaria. Educere. Artículos arbitrados. Año 8 No 25 julio-agosto- setiembre de 2004. 321-327.
- Carlino, P. (2008) La escritura en la investigación. Documento de Trabajo No 19. Conferencia pronunciada el 12 de noviembre de 2005 en el ámbito del Seminario permanente de investigación de la Maestría en Educación de la UdeSA.
- Cataldi, Z. y Lage, F. (2004). Diseño y organización de tesis. Bs. As.: Nueva Librería.
- Cohen N., Piovani J. I. (2008) La metodología de la investigación en debate. Editorial Univ. Nacional de la Plata.
- Eco, U. (1998) Cómo se hace una tesis: técnicas y procedimientos de estudio, investigación y escritura. Barcelona : Gedisa, 267 páginas
- Hernández Sampieri, R. y Otros. (2014- 6ta. Ed.). Metodología de la investigación. México: Mc Graw Hill.
- Marradi, A:, Archeri, N. y Piovani, H. (2007) Metodología de las Ciencias Sociales. Emecé.
- Pérez Serrano, G. (2002) Investigación cualitativa. II Retos e interrogantes: Técnicas y análisis de datos. Editorial la Muralla. Sabino, C. A. (1996) El proceso de investigación. 3a. ed. Buenos Aires: Lumen – Hvmanitas.
- Sautu, R. (2007) Práctica de la investigación cuantitativa y cualitativa articulación entre la teoría los métodos y las técnicas. Editorial Lumiere:
- Sierra Bravo R. (2005).Tesis Doctorales y Trabajos de investigación Científica. Thomson. Tamayo y Tamayo, M. (2002). El proceso de investigación científica. México: Limusa.
- Taylor, S.J.; Bogdan; R. (1987) Introducción a los métodos cualitativos de investigación: la búsqueda de significados. Barcelona: Paidós, 343 páginas.
- Vasilachis de G, I. Ameigeiras Aldo R., Chernobilsky Lilia B., Gimenez Beliveau, V. (2006) Estrategias de investigación cualitativa. Editorial Gedisa

Calle 50 y 120 | C.P.1900 | La Plata | Buenos Aires | República Argentina
Tel.: (54) 221-427-7270/71| www.info.unlp.edu.ar