Ingeniería en Sistemas de Información
Ingeniería de Software
Planificación Ciclo lectivo 2022

1. Datos administrativos de la asignatura			
Nivel en la carrera	Cuarto	Duración	96 horas
Bloque curricular:	☐ Programación ☐ Computación ☑ Sistemas de Información ☐ Gestión Ingenieril ☐ Modelos ☐ Asignatura Electiva		
Carga horaria presencial semanal:	6 horas	Carga Horaria total:	6 horas
Carga horaria no presencial semanal (si correspondiese)	N/A	% horas no presenciales (si correspondiese)	N/A

2. Plantel docente de la asignatura			
Docentes Titular Adjunto	 Judith Meles Laura Covaro	Dedicación:	
Jefe de Trabajos Prácticos	Cecilia Massano Gerardo Boiero	Dedicación:	
Auxiliar de 1ra.	Joaquín Robles Mickaela Crespo	Dedicación:	

3. Presentación, Fundamentación

4. Relación de la asignatura con las competencias de egreso de la carrera

Competencias específicas de la carrera (CE): Nivel de tributación	Competencias genéricas tecnológicas (CT) : Nivel de tributación	Competencias genéricas sociales, políticas y actitudinales (CS) : Nivel de tributación
CE1.1: Nivel	CT1: Nivel	CS6: Nivel
CE1.2: Nivel	CT2: Nivel	CS7: Nivel
CE1.3: Nivel	CT3: Nivel	CS8: Nivel
CE2.1: Nivel	CT4: Nivel	CS9: Nivel
CE3.1: Nivel	CT5: Nivel	CS10: Nivel
CE4.1: Nivel		
CE5.1: Nivel		

Competencias específicas de la carrera (CE)
1.1. Especificar, proyectar y desarrollar sistemas de información.
1.2. Especificar, proyectar y desarrollar sistemas de comunicación de datos.
1.3. Especificar, proyectar y desarrollar software.
2.1. Proyectar y dirigir lo referido a seguridad informática.
3.1. Establecer métricas y normas de calidad de software.
4.1. Certificar el funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.
5.1. Dirigir y controlar la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.

Competencias genéricas tecnológicas (CT):
1. Identificar, formular y resolver problemas de ingeniería.
2. Concebir, diseñar y desarrollar proyectos de ingeniería.
3. Gestionar, planificar, ejecutar y controlar proyectos de ingeniería.
4. Utilizar de manera efectiva las técnicas y herramientas de aplicación en la ingeniería.
5. Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.

Competencias genéricas sociales, políticas y actitudinales (CS)
6. Desempeñarse de manera efectiva en equipos de trabajo.
7. Comunicarse con efectividad.
8. Actuar con ética, responsabilidad profesional y compromiso social, considerando el impacto económico, social y ambiental de su actividad en el contexto local y global.
9. Aprender en forma continua y autónoma.
10. Actuar con espíritu emprendedor.

Fuente: PROPUESTA DE ESTÁNDARES DE SEGUNDA GENERACIÓN PARA LA ACREDITACIÓN DE CARRERAS DE INGENIERÍA EN LA REPÚBLICA ARGENTINA (Libro Rojo de CONFEDI)

5. Propósito

Brindar a los estudiantes técnicas y herramientas que permitirán la resolución de problemas relacionados con la Ingeniería de Software y las disciplinas que la integran, desde la aplicación de conceptos teóricos y mediante el desarrollo de competencias técnicas."

6. Objetivos establecidos en el DC

- Conocer los componentes de un proyecto de ingeniería de software.
- Conocer los estándares asociados a la calidad del proceso de desarrollo de software y de productos de software.
- Conocer los componentes de un plan de aseguramiento de calidad.
- Emplear las métricas que se aplican al desarrollo de software.
- Aplicar los elementos de un proceso de prueba ("testing").
- Diseñar un plan de prueba unitario y de integración.

7. Resultados de aprendizaje

- Identificar los componentes de un proyecto de desarrollo de software en el contexto de la gestión de proyectos basado en procesos definidos.
- Distinguir entre los diferentes ciclos de vida para el desarrollo de software, sus ventajas y desventajas en el marco de la Administración de Proyectos de Software.
- Descubrir herramientas para la gestión lean ágil de productos de software con la finalidad de aplicarlas en proyectos.
- Aplicar un framework de gestión ágil de proyectos con la finalidad de incorporar las prácticas en la gestión de proyectos de software.
- Calcular métricas para obtener visibilidad en el contexto de proyectos de desarrollo de software.
- Utilizar frameworks Lean-Ágiles para gestionar productos de software acordes a las expectativas de los involucrados.
- Reconocer la importancia de la disciplina de Gestión de Configuración de Software para construir productos de software de calidad.
- Citar conceptos relacionados con continuous integration, continuous delivery & continuous deployment tomando en cuenta su utilidad para obtener software de calidad.
- Analizar los principales modelos de calidad de software existentes en el mercado para poder evaluar cuál es el más adecuado para aplicar en un contexto particular.
- Identificar técnicas y herramientas para hacer aseguramiento de calidad de software en los proyectos de desarrollo de software.
- Descubrir la importancia de la prueba del software para controlar la calidad del producto construido.
- Utilizar técnicas (auditorías, revisión e inspecciones de software) relacionadas con el aseguramiento de la calidad del proceso y del producto con la finalidad de entregar un producto de software de calidad.

8. Asignaturas correlativas previas

Para cursar debe tener cursada:

- Probabilidad y Estadística
- Diseño
- Gestión de Datos

Para cursar debe tener aprobada:

- Análisis de Sistemas
- Sintaxis y Semántica del Lenguaje
- Paradigmas de Programación

Para rendir debe tener aprobada:

- Análisis de Sistemas
- Sintaxis y Semántica del Lenguaje
- Paradigmas de Programación

9. Asignaturas correlativas posteriores

Indicar las asignaturas correlativas posteriores:

- Proyecto Final

10. Contenidos Mínimos

- Componentes de un proyecto de Sistemas de Información.
- Gestión de Configuración de Software.
- Modelos de Calidad de Software. Aseguramiento de la Calidad.
- Métricas de Software.
- Auditoría y Peritaje

11. Programa analítico, Unidades temáticas

Unidad Nro. 1: Ingeniería de Software en Contexto

Resultados de Aprendizaje:

Al finalizar esta unidad se espera que el estudiante sea capaz de:

- Identificar los componentes de un proyecto de desarrollo de software en el contexto de la gestión de proyectos basado en procesos definidos.
- Explicar las razones que ocasionaron la llamada “crisis del software, desde diferentes puntos de vista conceptuales, aportando una conclusión.

- Distinguir entre los diferentes ciclos de vida para el desarrollo de software, sus ventajas y desventajas en el marco de la Administración de Proyectos de Software.
- Demostrar comprensión de la relación existente entre el Proceso, el Proyecto y el Producto en el contexto del desarrollo de software.

Contenidos:

- Introducción a la Ingeniería del Software. ¿Qué es?
- Estado Actual y Antecedentes. La Crisis del Software.
- Disciplinas que conforman la Ingeniería de Software.
- Ejemplos de grandes proyectos de software fallidos y exitosos.
- Ciclos de vida (Modelos de Proceso) y su influencia en la Administración de Proyectos de Software.
- Procesos de Desarrollo Empíricos vs. Definidos.
- Ciclos de vida (Modelos de Proceso) y Procesos de Desarrollo de Software
- Ventajas y desventajas de c/u de los ciclos de vida. Criterios para elección de ciclos de vida en función de las necesidades del proyecto y las características del producto.
- Componentes de un Proyecto de Sistemas de Información.
- Vinculo proceso-proyecto-producto en la gestión de un proyecto de desarrollo de software.

Bibliografía recomendada para la unidad 1:

- Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. (7ma Ed.). Mc Graw - Hill Interamericana. Capítulo 1, 24.
- Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). Mexico: Addison- Wesley. Capítulo 1, 22, 23.
- McConnell, Steve - SOFTWARE ESTIMATION: DEMYSTIFYING THE BLACK ART (Editorial Microsoft Press – Año 2006). Partes I y II
- Mc Connell, Steve. (1996). Desarrollo y Gestión de Proyectos Informáticos. Editorial McGraw Hill. Capítulo 7.
- SEBOK V 1.9.1 (Software Engineering Body of Knowledge)- IEEE 2018 - https://www.sebokwiki.org/wiki/Download_SEBoK_PDF
- Brooks, Frederick The mythical man-month (anniversary ed.) (1995) Addison-Wesley Longman Publishing Co., Inc.- Capítulos 1 al 3.

Papers:

- **Orphans Preferred (http://www.stevemcconnell.com/psd/07-orphanspreferred.htm)**
- **No Silver Bullet (http://www.virtualschool.edu/mon/SoftwareEngineering/BrooksNoSilverBullet.html)**
- **Software's Ten Essentials (http://www.stevemcconnell.com/ieeesoftware/10Essentials.pdf)**
- **http://martinfowler.com/articles/newMethodology.html Fowler, Martin** – The new methodology

Actividades a desarrollar:

Las actividades a desarrollar en la unidad incluyen clases expositiva participativas para el abordaje de los conceptos teóricos.

La cátedra tiene un curso en el aula virtual donde los estudiantes tienen acceso a todo el material bibliográfico, copias de presentaciones de clases, clases del 2021 grabadas, además de un foro disponible para atención de consultas de los estudiantes.

Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software

Resultados de Aprendizaje:

Al finalizar esta unidad se espera que el estudiante sea capaz de:

- Interpretar la filosofía lean y la filosofía ágil para poder determinar que enfoque es mejor en cada contexto particular de desarrollo de software.
- Descubrir herramientas para la gestión lean ágil de productos de software con la finalidad de aplicarlas en proyectos.
- Aplicar un framework de gestión ágil de proyectos con la finalidad de incorporar las prácticas en la gestión de proyectos de software.
- Calcular métricas para obtener visibilidad en el contexto de proyectos de desarrollo de software.
- Analizar comparativamente los enfoques de gestión tradicionales basados en procesos definidos con los enfoques de gestión basados en procesos empíricos con la meta de elegir el más adecuado a un contexto particular.
- Utilizar frameworks Lean-Ágiles para gestionar productos de software acordes a las expectativas de los involucrados.
- Ejercitar la técnica de user stories para la identificación de requerimientos en el contexto de proyectos de desarrollo ágiles

Contenidos

- Manifiesto Ágil/Filosofía Lean
- Requerimientos en ambientes lean ágil
- Introducción al Desarrollo Ágil.
- Requerimientos en ambientes ágiles - User Stories
- Estimaciones en ambientes ágiles
- Frameworks de SCRUM a nivel equipo y escala
- Métricas Ágiles
- Gestión de Productos de Software – Planificación de Productos – Herramientas para Definición de Productos de Software
    - Lean UX
    - Design Thinking

Bibliografía recomendada para la unidad 2:

- Cohn, Mike (2004) User Stories Applied – Editorial Addison Wesley. Capítulo 16.
- Gothelf, Jeff – Lean UX: Applying Lean Principles to Improve User Experience – Editorial O'Reilly, 2013.
- Schneider Jonny (2017). Understanding Design Thinking, Lean and Agile – Editorial O'Reilly.
- The Scrum Guide 2020 - http://www.scrumguides.org/download.html
- The Nexus Scrum Guide 2020 - https://www.scrum.org/resources/nexus-guide
- Leffingwell, Dean and Behrens Pete (2009). A user story primer Whitepaper.
- Manifiesto Ágil http://agilemanifesto.org/iso/es/
- http://pgpubu.blogspot.com.ar/2007/01/tcnica-de-estimacin-wideband-delphi.html
- http://people10.com/blog/software-sizing-for-agile-transformation

Actividades a desarrollar:

Las actividades a desarrollar en la unidad incluyen clases expositiva participativas para el abordaje de los conceptos teóricos. Además se abordarán algunos conceptos de la unidad con trabajos conceptuales grupales, que implican la presentación por parte de cada grupo al grupo clase.

Para las clases prácticas se utilizan como estrategias el aprendizaje basado en problemas y el estudio de casos. En la primera clase de cada tema los docentes muestran un caso con una solución propuesta, para que se tome como referencia. Luego, se trabaja en grupos pequeños y algún representante de un grupo presenta al grupo clase una solución al problema y se discute sobre la misma, obteniendo las conclusiones y atendiendo las dudas.'

El tipo de formación práctica es de ejercicios prácticos que se trabajan en clase con el objetivo de terminarlos y entregarlos durante la clase.

La cátedra tiene un curso en el aula virtual donde los estudiantes tienen acceso a todo el material bibliográfico, copias de presentaciones de clases, clases del 2021 grabadas, además de un foro disponible para atención de consultas de los estudiantes.

Unidad Nro. 3: Gestión del Software como producto

Resultados de Aprendizaje:

Al finalizar esta unidad se espera que el estudiante sea capaz de:

- Reconocer la importancia de la disciplina de Gestión de Configuración de Software para construir productos de software de calidad.
- Describir las actividades principales de la disciplina Gestión de Configuración de

Software considerando como contexto a los proyectos de desarrollo de software.
- Comparar diferentes herramientas utilizadas para la Gestión de Configuración de Software para discutir su uso para el desarrollo de software.
- Citar conceptos relacionados con continuous integration, continuous delivery & continuous deployment tomando en cuenta su utilidad para obtener software de calidad.

Contenidos.
- Conceptos Introductorias de la Gestión de Configuración.
- Versiones, variantes, release.
- Planificación de la Gestión de Configuración de Software.
- Actividades relacionadas con la Gestión de Configuración.
- El rol de las líneas base y su administración.
- Elementos de configuración del Software.
- Identificación de ítems de configuración en la Configuración de un software.
- Gestión de Configuración en ambientes ágiles
- Continuous Integration
- Continuous Delivery
- Continuous deployment - Estrategias de deployments - Canary Deployments- Blue/Green Deployment

Bibliografía recomendada para la unidad 3:

- Bersoff, Edgard – Elements of Software Configuration Management – Sitio: http://portal.acm.org
- Software Program Manager Network (1998) The Little Book of Software Configuration Management. AirLie Software Council- Sitio: http://www.spmn.com
- Rossel Sander. (2017). Continuous Integration, Delivery and Deployment, Editorial Packt
- M. Shahin, M. Ali Babar, and L. Zhu, (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices", IEEE Access.
- http://www.scmpatterns.com/pubs/hass_sidebar.html - Agile SCM
- http://www.scmpatterns.com/pubs/crossroads-mirror/agileoct03.pdf
- https://www.cmcrossroads.com/article/defining-agile-scm-past-present-future-2008?page=0%2C1
- https://www.atlassian.com/continuous-delivery/ci-vs-ci-vs-cd

Actividades a desarrollar:

Las actividades a desarrollar en la unidad incluyen clases expositiva participativas para el abordaje de los conceptos teóricos. Además se abordarán algunos conceptos de la unidad con trabajos conceptuales grupales, que implican la presentación por parte de cada grupo al grupo clase.

Para las clases prácticas se utilizan como estrategias el aprendizaje basado en problemas y el estudio de casos. En la primera clase de cada tema los docentes muestran un caso con una solución propuesta, para que se tome como referencia. Luego, se trabaja en grupos pequeños y algún representante de un grupo presenta al grupo clase una solución al problema y se discute sobre la misma, obteniendo las conclusiones y atendiendo las dudas.'

El tipo de formación práctica es de ejercicios prácticos que se trabajan en clase con el objetivo de terminarlos y entregarlos durante la clase.

La cátedra tiene un curso en el aula virtual donde los estudiantes tienen acceso a todo el material bibliográfico, copias de presentaciones de clases, clases del 2021 grabadas, además de un foro disponible para atención de consultas de los estudiantes.

Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto

Resultados de Aprendizaje:

Al finalizar esta unidad se espera que el estudiante sea capaz de:

- Descubrir las principales tendencias respecto a la calidad para incorporarlas al proceso de creación de software.
- Analizar los principales modelos de calidad de software existentes en el mercado para poder evaluar cuál es el más adecuado para aplicar en un contexto particular.
- Identificar técnicas y herramientas para hacer aseguramiento de calidad de software en los proyectos de desarrollo de software.
- Descubrir la importancia de la prueba del software para controlar la calidad del producto construido.
- Utilizar técnicas (auditorías, revisión e inspecciones de software) relacionadas con el aseguramiento de la calidad del proceso y del producto con la finalidad de entregar un producto de software de calidad.
- Plantear actividades relacionadas al aseguramiento de calidad de software e insertarlas en el contexto de un proyecto de desarrollo.

Contenidos

- Conceptos generales sobre calidad.
- Importancia de trabajar para y con Calidad. Ventajas y Desventajas.
- Actividades relacionadas con el Aseguramiento de la Calidad del Software.
- Principales Modelos de Calidad existentes (CMMI – SPICE – ISO) y sus métodos de evaluación.
- Lineamientos para la implementación de modelos de calidad en las organizaciones.

- Diferentes tipos de Auditorias: Auditorías de Proyecto y Auditorías al Grupo de Calidad.
- Proceso de Auditorías: Responsabilidades. Preparación y ejecución. Reporte y seguimiento.
- Calidad de Producto: Planificación de pruebas para el software- Niveles y tipos de pruebas para el software. Técnicas y herramientas para probar software. Técnicas y Herramientas para la realización de revisiones técnicas del software.
- Testing en ambientes Ágiles.
- Mejora continua de procesos con Kanban

Bibliografía recomendada para la unidad 4:

- Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). Mexico: Addison- Wesley. Capítulo 24 y 26
- Myers, Glenford (2011). The art of software testing- 3rd Edition. Editorial Wiley. / El arte de Probar el Software (1983). Editorial El Ateneo. Capítulos 2 al 6
- Crispin, Lisa & Gregory Janetm(2008) Agile Testing – A Practical Guide for Testing and Agile Teams. Editorial O' Really Media.
- Kniber Henric (2011). Lean from the trenches – Un example of Kanban for large software project. Editado por Key Keppler.
- Anderson, David J.(2011). Kanban. Editorial Blue Hole.
- Anderson, David J. & Carmichael, Andy (2016). Kanban Esencial Condensado Editorial LeanKanban University.
- IEEE STD 1028-1997 STANDARD FOR SOFTWARE REVIEWS
- IEEE STD 1012-1998 (REVISION OF IEEE STD 1012-1986) IEEE STANDARD FOR SOFTWARE VERIFICATION AND VALIDATION
- HTTP://TESTOBSESSED.COM/WP-CONTENT/UPLOADS/2011/04/AGILETESTINGOVERVIEW.PDF
- HTTP://WWW.AMBYSOFT.COM/ESSAYS/AGILETESTING.HTML

Actividades a desarrollar:

Las actividades a desarrollar en la unidad incluyen clases expositiva participativas para el abordaje de los conceptos teóricos. Además se abordarán algunos conceptos de la unidad con trabajos conceptuales grupales, que implican la presentación por parte de cada grupo al grupo clase.

Para las clases prácticas se utilizan como estrategias el aprendizaje basado en problemas y el estudio de casos. En la primera clase de cada tema los docentes muestran un caso con una solución propuesta, para que se tome como referencia. Luego, se trabaja en grupos pequeños y algún representante de un grupo presenta al grupo clase una solución al problema y se discute sobre la misma, obteniendo las conclusiones y atendiendo las dudas.'

El tipo de formación práctica es de ejercicios prácticos que se trabajan en clase con el objetivo de terminarlos y entregarlos durante la clase.

La cátedra tiene un curso en el aula virtual donde los estudiantes tienen acceso a todo el material bibliográfico, copias de presentaciones de clases, clases del 2021 grabadas, además de un foro disponible para atención de consultas de los estudiantes.

Carga horaria por unidad:

Unidad 1	12 horas
Unidad 2	40 horas
Unidad 3	8 horas
Unidad 4	28 horas
Evaluaciones	8 horas

Carga horaria por tipo de formación práctica

La carga horaria afectada a la formación práctica se corresponde con el 60 % de las horas totales de la asignatura, es decir 58 horas.

Bibliografía Obligatoria:

- Anderson, David J.(2011). Kanban. Editorial Blue Hole.
- Anderson, David J. & Carmichael, Andy (2016). Kanban Esencial Condensado Editorial LeanKanban University.
- Crispin, Lisa & Gregory Janetm(2008) Agile Testing – A Practical Guide for Testing and Agile Teams. Editorial O' Really Media.
- McConnell, Steve - SOFTWARE ESTIMATION: DEMYSTIFYING THE BLACK ART (Editorial Microsoft Press – Año 2006).
- Myers, Glenford (2011). The art of software testing- 3rd Edition. Editorial Wiley. / El arte de Probar el Software (1983). Editorial El Ateneo.
- Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. (7ma Ed.). Mc Graw - Hill Interamericana.
- Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). Mexico: Addison- Wesley.

Bibliografía optativa y otros materiales a utilizar en la asignatura:

- Cohn, Mike (2004) User Stories Applied – Editorial Addison Wesley.
- Cohn, Mike (2006). Agile Estimation and Planning – Editorial Prentice Hall.
- M. Shahin, M. Ali Babar, and L. Zhu, (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices", IEEE Access.
- Kniber Henric (2011). Lean from the trenches – Un example of Kanban for large software project. Editado por Key Keppler.
- Rossel Sander. (2017). Continuous Integration, Delivery and Deployment, Editorial Packt

- Schneider Jonny (2017). Understanding Design Thinking, Lean and Agile – Editorial O'Reilly.
- Mc Connell, Steve. (1996). Desarrollo y Gestión de Proyectos Informáticos. Editorial McGraw Hill.
- No Silver Bullet (http://www.virtualschool.edu/mon/SoftwareEngineering/BrooksNoSilverBullet.html)
- Leffingwell, Dean and Behrens Pete (2009). A user story primer Whitepaper.
- Manifiesto Ágil http://agilemanifesto.org/iso/es/
- The Scrum Guide 2020 - http://www.scrumguides.org/download.html
- The Nexus Scrum Guide 2020 - https://www.scrum.org/resources/nexus-guide
- http://pgpubu.blogspot.com.ar/2007/01/tcnica-de-estimacin-wideband-delphi.html
- http://people10.com/blog/software-sizing-for-agile-transformation
- Bersoff, Edgard – Elements of Software Configuration Management – Sitio: http://portal.acm.org
- Software Program Manager Network (1998) The Little Book of Software Configuration Management. AirLie Software Council- Sitio: http://www.spmn.com
- Gothelf, Jeff – Lean UX: Applying Lean Principles to Improve User Experience – Editorial O'Reilly, 2013
- Brooks, Frederick The mythical man-month (anniversary ed.) (1995) Addison-Wesley Longman Publishing Co., Inc.
- CMMI para Desarrollo en Español: http://cmmiinstitute.com/assets/Spanish%20Technical%20Report%20CMMI%20V%201%203.pdf
- SPICE Project, Consolidated product. Software Process Assessment – Part 1: Concepts and introductory guide. Version 1.00. Site de SPICE: www.esi.es/Projects/SPICE
- McFeeley, Bob - IDEAL: A User Guide for Software Process Improvement – CMU/SEI-96-HB-001. www.sei.cmu.edu
- Sitio de la IEEE: http://www.ieee.org
- IEEE Std 730 Standard for Software Quality Assurance Plans
- IEEE Std 1028-1997 Standard for Software Reviews
- IEEE Std 1012-1998 (Revision of IEEE Std 1012-1986)
- IEEE Standard for Software Verification and Validation
- SEBOK V 1.9.1 (Software Engineering Body of Knowledge)- IEEE 2018 - https://www.sebokwiki.org/wiki/Download_SEBoK_PDF

12. Metodología de enseñanza

Destacando el hecho que el currículo no solo se manifiesta en la especificación de una serie de contenidos en un programa, sino por el contrario, abarca cuestiones mucho más profundas tales como: bibliografía, priorización de algunos contenidos sobre otros, proceso de enseñanza – aprendizaje, formas de evaluación, entre otras; es que se considera importante poner de

manifiesto algunos de estos aspectos con el propósito de mejorar el nivel académico y fomentar la integración de la cátedra, sin interferir, por supuesto, en la libertad de cada uno de los docentes que la integren.

La selección de los contenidos incluidos en el programa se realizó considerando la integración de esta nueva asignatura al resto de las asignaturas de la carrera, lo que fundamenta en gran medida la priorización y el nivel de profundidad elegido para cada tema.

Dentro de las cuestiones que se expondrán para el desarrollo de la Metodología se tendrán en cuenta los siguientes aspectos:

Dictado de la materia: el contenido temático está organizado lógicamente y situado coherentemente según su grado de dificultad de manera que permita al estudiante ir asimilando los contenidos propios de la materia en forma gradual y a la vez integrar los contenidos de otras asignaturas.

Para el desarrollo del programa se considerará un sistema de clases que combine: clases explicativas, con clases prácticas tipo taller, clases invertidas, gamification y el uso del herramientas aplicadas para el desarrollo de software.

- A través de estas diferentes formas organizativas de la enseñanza se proponen los siguientes objetivos educativos: Transmitir los conocimientos a través de un proceso de enseñanza-aprendizaje que permita la apropiación de los contenidos.
- Desarrollar el hábito de la lectura, el análisis y la interpretación de textos, invitando a los alumnos a trabajar con las fuentes bibliográficas originales, posibilitando que elaboren sus propias interpretaciones y realicen sus propias conclusiones.
- Promover el espíritu investigativo para buscar siempre la verdad auténtica y la rigurosidad de la ciencia en la búsqueda de las soluciones a las situaciones de aprendizaje que se propongan.
- Valorar el uso de bibliografía como fuente original de los conceptos desarrollados en la asignatura.
- Fomentar la habilidad para aplicar los conocimientos adquiridos a situaciones concretas.

Además la catedra definió la cátedra definió lo siguiente:

- Un espacio en la UV para toda la cátedra, allí se compartirá todo el material necesario, se plantearán tareas para trabajos prácticos. Cada Estudiante se matricula en el curso que está inscripto.
- Quedan a disposición de los estudiantes los links de todas las clases grabadas del ciclo lectivo 2021, que están subidas en el canal de YouTube de la cátedra.

Respecto a las estrategias de enseñanza se utilizarán tanto clases expositivas, como clases invertidas, así como también actividades con gamification. Trabajos grupales con Aprendizaje Basado en Problemas, Estudio de Casos, y técnicas de exposición de los trabajos realizados. También se utilizarán recursos como mapas mentales y mapas conceptuales, cuadros sinópticos y comparativos para afianzar los conceptos abordados en las clases y para las evaluaciones.

13. Recomendaciones para el estudio

Las clases teóricas plantean y desarrollan los conceptos y su aplicación a los casos prácticos utilizando ejemplos; las clases prácticas dan continuidad al aprendizaje abordando los temas y afianzando los saberes privilegiando el "hacer". Dada esta estrategia las recomendaciones son:

- Asistir a las clases tanto conceptuales como de aplicación con el material que se aconsejó analizado.
- Hacer al menos los ejercicios que la cátedra pone a disposición.
- Evaluar individualmente si requieren práctica adicional, la que está disponible en el aula virtual de la cátedra.
- No quedarse con dudas sobre ningún tema, consultar a los docentes en el momento de la clase o solicitar una clase de consulta del modo que se explica más adelante en este documento.
- Participar en las actividades grupales propuestas, activamente.
- Utilizar las soluciones propuestas para contrastarlas con las propias, luego de resolver los ejercicios.
- Estudiar para los parciales utilizando la bibliografía recomendada.

14. Metodología de evaluación

Para regularizar, el estudiante deberá:

- Rendir y aprobar 2 (dos) parciales teóricos y 2 (dos) parciales prácticos.
- Puede acceder a 1 (un) recuperatorio de cada uno de los dos parciales, pero sólo uno de ellos. Los parciales recuperatorios se tomarán a fin del cuatrimestre.
- En caso de recuperar se considera la mejor nota obtenida por el estudiante.
- Cumplir con las condiciones explicadas en el apartado anterior, respecto de los trabajos grupales.

La nota mínima de aprobación es un 4 (cuatro),

La escala de notas para aprobación de parciales es la siguiente:

Nota	Porcentaje	Situación
1		No aprueba
2		No aprueba
3		No aprueba
4	55 % - 57 %	Aprueba
5	58% - 59 %	Aprueba
6	60 % - 68 %	Aprueba
7	69 % - 77%	Aprueba
8	78% - 86%	Aprueba
9	87% - 95 %	Aprueba
10	96% - 100 %	Aprueba

Forma de registrar las notas en la Autogestión Académica:

Evaluación	Etiqueta en Autogestión
Primer Parcial Teórico	1er. Teórico
Primer Parcial Práctico	1er. Práctico
Segundo Parcial Teórico	2do. Teórico
Segundo Parcial Práctico	2do. Práctico
Promedio de los TP's	1er. Integrador
Promedio de los Trabajos Conceptuales	3er. Teórico
Recuperatorio Teórico	1er. Recuperatorio
Recuperatorio Práctico	2do. Recuperatorio
Nota de Aprobación Directa	Nota Final

15. Modalidad de examen final

De no obtener aprobación directa, el estudiante deberá aprobar el examen final.

Al momento de la inscripción al examen final, el sistema de inscripción la asigna aleatoriamente un tema, basado en los contenidos de las unidades temáticas de la materia. Este tema será el primer tema que el estudiante exponga en su coloquio, de no alcanzar nivel satisfactorio en su exposición, el examen dará por finalizado con la no aprobación del estudiante. En caso contrario los docentes le asignarán dos temas más para que el estudiante desarrolle. Finalizado el coloquio se le informará la nota.

En esta instancia se evaluarán todos los contenidos del *último programa vigente* para la asignatura.

Se evaluarán aspectos conceptuales de la materia.

El examen final se aprueba con nota 6 (seis) o superior, correspondiendo al 60 % de los contenidos evaluados.

La cátedra tomará los exámenes finales en forma conjunta para todos los estudiantes, esto permitirá la nivelación e integración de todos los cursos que la conforman.

Escala de Notas:

Nota	Porcentaje	Situación
1		Insuficiente
2		Insuficiente
3		Insuficiente
4		Insuficiente
5		Insuficiente
6	60 % - 68 %	Aprobado
7	69 % - 77%	Bueno
8	78% - 86%	Muy Bueno
9	87% - 95 %	Distinguido
10	96% - 100 %	Sobresaliente

16. Cronograma de clases/trabajos prácticos/exámenes (tentativo)

Semana	Clase 1	Clase 2
14/03	Presentación de la materia, de los docentes Introducción a la Ingeniería de Software	Componentes de un proyecto de software
21/03	Teórico Práctico de Filosofía Ágil/ Manifiesto Ágil	Requerimientos en Ambientes Ágiles - User Stories
28/03	Práctico de User Stories	Gestión de Productos y Estimaciones de Software
04/04	Práctico de User Stories con Estimaciones y MVP	Administración de Configuración de Software (Clase invertida basada en el video del 4K4)
11/04	Administración de Configuración de Software	SCRUM (clase invertida con lo básico de SCRUM que está en la guía)
18/04	Dinámicas de SCRUM- Explicación del práctico de implementación de user stories	Teórico de Framework para escalar Scrum
25/04	Práctico de Scrum	Clase de Consulta
03/05	**Primer Parcial Teórico y Primer Parcial Práctico**	
05/05	Testing de Caja Negra (Clase invertida con video de testing)	Overview de Testing
12/05	Testing Caja Blanca (Clase invertida con video de testing)	Testing ágil en contexto
19/05	Corrección de las implementaciones de las user stories	Aseguramiento de Calidad de Proceso y de Producto
26/05	Práctico de ejecución de testing de caja negra con la us implementada.	Filosofía Lean y Kanban
02/06	TP13 – No Entregable. Práctico de Caja Blanca de US a probar de otro grupo.	Clase de Consulta
07/06	**Segundo Parcial Teórico y Segundo Parcial Práctico**	
09/06	Práctico de Publicidad en Instagram con herramientas de DT	Métricas tradicionales, lean y agile
16/06	Revisiones técnicas (clase invertida)	Comparación de enfoques tradicional, lean y agile
23/06	Prácticas Continuas	Retrospectivas
28/06	**Recuperatorios Teóricos y Prácticos**	
01/07	Regularización	TC3: Pecha Kucha (Testing Agile)

17. Recursos necesarios

- TBD

18. Función Docencia

- TBD

19. Reuniones de asignatura y área

- TBD

20. Atención y orientación de estudiantes

- La cátedra tiene planificada una clase de consulta la clase previa a cada uno de los parciales, inclusive los recuperatorios.
- Los horarios de consulta deberán convenirlos en cada curso con sus docentes.
- Las direcciones de correo electrónico de cada docente están publicadas en la UV de la cátedra, para que los estudiantes se comuniquen cuando así lo requieran.
- A todo efecto la cátedra dispone de la siguiente dirección de correo para que los alumnos se puedan comunicar directamente con el Coordinador de Cátedra: **jmeles@gmail.com**.