



Gestión de desarrollo de software

TUP MARCOS JUÁREZ

Planificación Ciclo lectivo 2025

Datos administrativos de la asignatura			
Carrera:	TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN		
Asignatura:	Gestión de desarrollo de software		
Ciclo Académico	2026		
Nivel de la carrera	Pre-Grado	Duración	2 años
Plan de Estudios	2024	Modalidad	Presencial
Horas reloj semanal:	4	Horas reloj total:	64
Docente	Ing. Gustavo San Felipe	Conf. Parcial	



Planificación de la Asignatura

FUNDAMENTACIÓN DE LA ASIGNATURA

Gestión de Desarrollos de Software, proporciona fundamentos para conocer y comprender cómo se gestionan en la actualidad proyectos informáticos, realizando un recorrido por las distintas etapas del diseño y modelización de datos y reconociendo las distintas características, funcionalidades, herramientas, momentos, etapas, costos, alcances y límites de un proyecto informático. El propósito fundamental de su enseñanza, es que los estudiantes tengan noción de la importancia que adquiere la gestión de proyectos hoy en día, adquiriendo el conocimiento de los distintos componentes de su entorno e interpreten las funcionalidades que determinan sus usos. El estudiante desarrollara capacidades para diseñar el alcance, limitantes, recursos a incluir y/o excluir, así como la reconocer los instrumentos y/o técnicas de medición, que permita cuantificar y cualificar el proyecto. Realizar diseños de documentos y generación de informes.

OBJETIVOS ESTABLECIDOS EN EL DISEÑO CURRICULAR

1. Utilizar herramientas para la correcta gestión de un proyecto de software, integrando y consolidando los conocimientos adquiridos.
2. Desarrollar habilidades relacionadas con la administración de tiempos y recursos.
3. Integrar el marco teórico establecido y los conocimientos previos, pasando por todas las etapas y roles en el desarrollo de un proyecto.



CONTENIDOS

Unidad 1 – Fundamentos y Propuesta de Proyecto

Software: **proceso** **y** **producto.**

- Introducción al ciclo de vida del software.
- Ingeniería de Requerimientos: identificación, análisis, especificación.
- Propuesta de proyecto: descripción, objetivos.
- Alcance: inclusiones y exclusiones.
- Registro de interesados (stakeholders).
- Criterios de aceptación, supuestos y restricciones del proyecto.

Unidad 2 – Planificación y Estimación

- Estimación de tiempos y costos en proyectos de TI.
- Métricas de software: líneas de código, puntos de función, velocidad ágil.
- Calendarización del proyecto.
- Cronograma de hitos y entregables.
- Definición de iteraciones del proyecto.

Unidad 3 – Ejecución y Seguimiento del Proyecto

- Artefactos técnicos requeridos: documentación técnica y código.
- Reportes, indicadores y estadísticas en proyectos de software.
- Control de avance del proyecto.
- Metodologías tradicionales y ágiles aplicadas al seguimiento.
- Gestión de riesgos y calidad.

Unidad 4 – Cierre del Proyecto y Documentación

- Preparación de manuales e instructivos.
- Presentación de entregables finales.
- Evaluación del cumplimiento de objetivos y criterios de aceptación.



- Cierre administrativo del proyecto.
- Presentación de resultados y lecciones aprendidas.



ESTRATEGIAS METODOLÓGICAS

El proceso de enseñanza y el proceso de aprendizaje se organizará alrededor de los siguientes supuestos básicos:

- En el aula, el estudiante es sujeto dinámico.
- El aprender es un proceso activo por parte del sujeto.
- El trabajo grupal influye en el desarrollo intelectual del que aprende.
- El aporte de distintos puntos de vista contribuye a la estructuración del pensamiento.

Se aprende mejor lo que interesa y se relaciona con la realidad vivenciada

Cuando los procesos de aprendizaje parten de problemas relacionados con las experiencias previas, cuando los resultados del aprendizaje son aplicables a problemas de la vida real; cuando, a través de un proceso gradual de interiorización y abstracción, con aclaraciones y reconstrucciones, se obtienen conceptos y operaciones, estamos hablando de aprendizaje significativo.

Las estrategias de desarrollo de la materia se centrarán fundamentalmente en el estudiante con una relación sinérgica entre teoría y práctica, evitando la dicotomía entre ambas acciones.

Se propone abordar cada tema desde planteamientos claros, conceptualización crítica y la participación continua de todos los actores áulicos en la construcción de conocimientos y habilidades.

Se trabajará continuamente con el planteo de casos y la resolución de situaciones problemáticas como medio de acercar al estudiante a la realidad que vivenciará como profesional.

Se pondrá énfasis en el trabajo en equipo, tendiente a constituir verdaderos equipos de trabajo

ACTIVIDADES FORMATIVAS PROPUESTAS

- Taller de identificación de requerimientos para un sistema simulado (ej: sistema de gestión académica).
 - Elaboración de la propuesta inicial del proyecto y registro de interesados.
 - Creación de un cronograma con herramientas (MS Project, GanttProject o Trello).
 - Taller de estimación de tiempos mediante Puntos de Función / Planning Poker.
 - Definición de iteraciones y entregables para el proyecto grupal.
 - Taller de construcción de artefactos (documentación técnica + primeros prototipos de código).
 - Uso de herramientas de seguimiento (Jira, Trello, GitHub Projects).
 - Elaboración de reportes de avance e indicadores.
 - Elaboración de manual de usuario y manual técnico del sistema desarrollado.
- Presentación final del **proyecto integrador grupal** (definición, planificación, ejecución y cierre).



EVALUACIÓN

Instrumentos: Trabajo Final Integrador.
Evaluación en proceso .
Cumplimiento de metas y tiempos planificados.

A) Aprobación No Directa (regularidad):

- **Asistencia** exigida por la Ordenanza N°1622.
- Rendir y **aprobar** dos **(2) exámenes parciales teórico-práctico** con nota superior o igual a seis **(6)**, teniendo la posibilidad de realizar un **(1) o más recuperatorios**, con idénticas condiciones de aprobación.
- **Aprobar los trabajos prácticos** que se propongan en su totalidad.
- **Presentación del trabajo práctico final integrador** (Individual / grupal).

B) Aprobación Directa:

- **Aprobar los exámenes parciales teórico-práctico con nota Promedio** superior o igual a ocho **(8)**.
En el caso en el que el estudiante no haya aprobado alguna de las instancias de evaluación parcial, tendrá al menos 1(una) instancia de recuperación lo cual deberá consignarse en la planificación de cátedra.

C) Vigencia:

- La regularidad tendrá validez por **todo el año lectivo siguiente al que regularizó**, si el estudiante cursó en el año 2024, mantendrán la regularidad hasta los turnos de febrero-marzo del 2026.

D) Examen Final:

- Para obtener la aprobación de la materia el estudiante deberá **aprobar el examen final** con nota superior o igual a **6** (seis).
- Si ha obtenido la **Aprobación Directa**, el alumno **deberá inscribirse** para poder registrar la nota.

→ Horario de consultas

Las clases de consultas se coordinan con los estudiantes según sus necesidades.

Se enfatiza el uso de los recursos de comunicación asincrónica (foros) y sincrónica (chat) que proporciona la plataforma virtual (Moodle).

Se planifican encuentros individuales o grupales por medio de videoconferencia por las diversas plataformas.



RECOMENDACIONES PARA EL ESTUDIO

- Lectura semanal de la bibliografía propuesta por la cátedra.
- Análisis y resolución modelo de los casos de estudios práctico propuestos en forma semanal.
- Confeccionar lista de preguntas, dudas y comentarios reflexivos sobre el material propuesto semanalmente.



CRONOGRAMA DE CLASES AÑO 2026					
CRONOGRAMA PROGRAMACIÓN III					
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN					
FECHA	Nº	UNIDAD	CONTENIDOS TEMÁTICOS	ESTRATEGIAS DE ENSEÑANZA	EVALUACIÓN
	1	Fundamentos y Propuesta de Proyecto	• Software: proceso y producto. • Introducción al ciclo de vida del software. • Ingeniería de Requerimientos: identificación, análisis, especificación. • Propuesta de proyecto: descripción, objetivos. • Alcance: inclusiones y exclusiones. • Criterios de aceptación, supuestos y restricciones del proyecto. • Registro de interesados (stakeholders).	Explicacion Teórico / Práctica Ejemplos en Pizarra / Proyección Diapositivas	



	2	Planificación y Estimación	• Estimación de tiempos y costos en proyectos de TI. • Métricas de software: líneas de código, puntos de función, velocidad ágil. • Calendarización del proyecto. • Cronograma de hitos y entregables. • Definición de iteraciones del proyecto.	Explicación Teórico / Práctica Ejemplos en Pizarra / Proyección Diapositivas	
	3	Ejecución y Seguimiento del Proyecto	• Artefactos técnicos requeridos: documentación técnica y código. • Reportes, indicadores y estadísticas en proyectos de software. • Control de avance del proyecto. • Metodologías tradicionales y ágiles aplicadas al seguimiento. • Gestión de riesgos y calidad.	Explicación Teórico / Práctica Ejemplos en Pizarra / Proyección Diapositivas	



	4	Cierre del Proyecto y Documentación	• Preparación de manuales e instructivos. • Presentación de entregables finales. • Evaluación del cumplimiento de objetivos y criterios de aceptación. • Cierre administrativo del proyecto. • Presentación de resultados y lecciones aprendidas. • Tendencias actuales: DevOps, CI/CD, gestión ágil a gran escala.	Explicación Teórico / Práctica Ejemplos en Pizarra / Proyección Diapositivas	
--	---	-------------------------------------	---	---	--



REFERENCIAS BIBLIOGRÁFICAS

A) Obligatoria o básica:

- *Sommerville, I. (2016). Ingeniería de Software. Pearson.*
- *Pressman, R. (2014). Ingeniería del Software: Un enfoque práctico. McGraw-Hill.*
- *Schwaber, K., & Sutherland, J. (2020). Scrum Guide.*

B) Complementaria:

- *Herramientas de gestión: Trello, Jira, GitHub Projects, MS Project / GanttProject.*
- *Control de versiones: GitHub / GitLab.*
- *Material de cátedra (guías de requerimientos, cronogramas, ejemplos de métricas).*
- *Casos de estudio de proyectos de software reales.*

Esta planificación es necesaria para poder certificar los servicios prestados.