	1. Reglas de juego para el curso:: Introducción a la Ingeniería de Software: Características del software de pequeña, mediana y gran envergadura. Problemas claves de desarrollo de software mediano y grande. La Ingeniería del Software. Actividades típicas de ingeniería: análisis, diseño, construcción, verificación, operación, mantenimiento, gestión.	T1: Introducción al Taller Programación (1): Programación por pares (1). Equipos: Introducción a los CVs y entrevistas La importancia de los respaldos. **Enunciar Tarea 1**: Base de Datos	
2	2. Diseño (1): Principios de diseño: Dividir para conquistar, acoplamiento, cohesión, Programación (2): Programación dirigida por contratos y programación dirigida por casos de prueba. Refactorización (1).	*Feriado* (1 de mayo)	T2: **Entrega de avance sobre Tarea 1**
3	3. Diseño (2): Diagramas de clase y diagramas EER. Codificación de un diagrama de clase. Verificación (1): Repaso de testing unitario. Barreras psicológicas(1).	4. Análisis (1): Requerimientos funcionales y no-funcionales (flexibilidad, desempeño). Diseño (3): Necesidad de arquitectura. Concepto de patrones. [El patrón arquitectónico zaguán/foyer] Patrón de capas: ejemplos SO, protocolos redes.	T3: **Entrega de Tarea 1**. JUnit. **Enunciado Tarea 2**: JUnit
4	5. Diseño(4): El patrón Fachada. Modelos ágiles: Características de los modelos ágiles. Ventajas, desventajas y riesgos resaltantes. Iteraciones, fases y dinámica de desarrollo.	6. Equipos (2): Fundamentos de la gestión de riesgos aplicada al desarrollo ágil de proyectos de software. Roles y responsabilidades, gestión de reuniones, agendas, minutas y calendarios, dinámica de comunicaciones (verbales, escritas y electrónicas) efectivas. Verificación (2): El plan de pruebas. Pruebas caja negra y caja blanca.+…. Programación (3):	T4: **Entrega de Tarea 2.** **Enunciado Tarea 3**: Patrón Fachada

		Refactorización de código (2)	
5	7. Diseño (5): El patrón Estrategia Equipos (3): Estimación (ad-hoc) de esfuerzo, y distribución y seguimiento de tareas, seguimiento del desarrollo, hojas de registro de trabajo. El uso de herramientas de apoyo para el seguimiento de defectos y la gestión de versiones, configuración y liberaciones.	8. Diseño (6): El patrón Decorador. Gestión de riesgos de diseño Programación (4): Refactorización de código (2). Estrategias de integración de código. Verificación (3): Pruebas de integración.	T5: Github
6	9. Examen 1	10. Diseño (7): Justificación del diseño: el rol de los requisitos en el diseño. El patrón Observador, Editorial-suscriptores. Programación (5): Refactorización de código (3)	T6: **Entrega de Tarea 3** Dinámica de Scrum. **Enunciado Tarea 4**: Patrones Estrategia y Decorador
7	11. Diseño (8): Principios de diseño: ocultamiento de información, separación de preocupaciones, encapsulamiento, Patrón MVC. ¿Qué requisitos llevan/se alejan de arquitectura de capas? Gestión de riesgos de diseño. Verificación (4)	12. Análisis (2): Casos de uso. Lectura y comprensión de requerimientos funcionales. Análisis de completitud, consistencia y factibilidad de requerimientos funcionales y no-funcionales. Importancia del caso de negocio y la gestión de riesgos de análisis. Verificación (5): Distinción entre validación y verificación. Barreras psicológicas (2)	T7: Herramienta de apoyo para el desarrollo de una interfaz simple (Spring)
8	13. Diseño (9): Diseño orientado por objetos y sus diferencias respecto a los paradigmas de diseño estructurado y diseño centrado en estructuras de datos.	14. Verificación (6): Revisiones e inspecciones. Ejercicio: [Revisión, evaluación y verificación de la documentación técnica del software y del proyecto]. Gestión de riesgo de pruebas.	T8: **Entrega de Tarea 4** Integración de equipos de pares . **Enunciado Tarea 5**: Interfaz

			en Spring + Integración
9	15. *[Buffer]* Verificación (): Ejercicio de inspección. Equipo (4): Dinámica de comunicaciones (verbales, escritas y electrónicas) efectivas , herramientas electrónicas de colaboración, escucha activa.	16. Equipo (5): El manejo de pluralidad, solución de conflictos.	T9:
10	*Feriado* (24 de junio)	17. Ética (1): [Introducción al marco legal que regula la ingeniería de software en Venezuela. La presión de la entrega versus los estándares de calidad profesional]	T10:
11	18. Ética (2): [Introducción a los códigos éticos de cuerpos nacionales e internacionales pertinentes como el Colegio de Ingenieros de Venezuela, ACM, IEEE-CS, IFIP. Disenso ético y la denuncia profesional. Prevención y solución de problemas de acoso y discriminación en equipos de desarrollo.]	19. Selección de herramientas	*Feriado* (5 de julio)
12	20. Examen 2	21. Equipo(): Análisis postmortem (proyecto y curso). Carrera: Vinculación de la asignatura con la carrera. Carreras afines a Ingeniería de Software..	T11: Entrega final Tarea 5.