

Metodologías de Desarrollo de Software: Fundamentos y Prácticas Modernas

Ingeniería | Ingeniería de sistemas | para estudiantes universitarios | 16 semanas

Descripción del Curso

Este curso ofrece una introducción integral a las metodologías de desarrollo de software, enfocándose en su aplicación en proyectos reales dentro del ámbito de la ingeniería de sistemas. Su propósito es dotar a los estudiantes universitarios de los conceptos, técnicas y herramientas necesarias para planificar, ejecutar y gestionar procesos de desarrollo de software con calidad y eficiencia.

Dirigido a estudiantes de ingeniería interesados en profundizar en los enfoques estructurados y ágiles para el desarrollo de software, el curso aborda desde los fundamentos teóricos hasta la aplicación práctica de metodologías contemporáneas. Se enfatiza la comprensión crítica y comparativa de diferentes modelos para seleccionar y adaptar la metodología adecuada según el contexto del proyecto.

El enfoque metodológico combina exposiciones teóricas, análisis de casos, actividades colaborativas y ejercicios prácticos que permiten a los estudiantes experimentar con metodologías tradicionales como el Modelo en Cascada y el Prototipado, así como metodologías ágiles como Scrum y Kanban. Al finalizar, los estudiantes serán capaces de diseñar, implementar y evaluar procesos de desarrollo de software que respondan a las necesidades y restricciones de proyectos reales.

Objetivos Generales

- Identificar y describir las principales metodologías de desarrollo de software, sus características, ventajas y desventajas.
- Aplicar metodologías tradicionales y ágiles en el diseño y gestión de proyectos de software, adaptando las prácticas a diferentes escenarios.
- Evaluar críticamente procesos y resultados en proyectos de desarrollo de software para proponer mejoras continuas.
- Integrar herramientas y técnicas de gestión de proyectos y calidad en el ciclo de vida del desarrollo de software.
- Comunicar efectivamente ideas y resultados relacionados con las metodologías de desarrollo de software en contextos académicos y profesionales.

Competencias

- Analizar y comparar diferentes metodologías de desarrollo de software para seleccionar la más adecuada según el contexto del proyecto.

- Diseñar planes de desarrollo utilizando metodologías tradicionales y ágiles, aplicando buenas prácticas de ingeniería de software.
- Implementar técnicas de gestión de proyectos y control de calidad en el ciclo de vida del desarrollo de software.
- Colaborar eficazmente en equipos multidisciplinarios aplicando métodos ágiles para el desarrollo iterativo e incremental de software.
- Evaluar riesgos, recursos y tiempos en proyectos de software para optimizar la entrega y satisfacción del cliente.

Requerimientos

- Conocimientos básicos de programación y fundamentos de ingeniería de software.
- Familiaridad con conceptos generales de gestión de proyectos.
- Acceso a un entorno computacional con herramientas para la gestión y desarrollo de software (por ejemplo, software de control de versiones, plataformas colaborativas).
- Competencias básicas en trabajo colaborativo y comunicación efectiva.

Unidades del Curso

Unidad 1: Introducción a las Metodologías de Desarrollo de Software

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de definir los conceptos fundamentales del desarrollo de software con precisión en un contexto académico.
- Al finalizar la unidad, el estudiante será capaz de describir la importancia y el propósito de las metodologías de desarrollo de software, ejemplificando su impacto en proyectos reales.
- Al finalizar la unidad, el estudiante será capaz de identificar las fases principales del ciclo de vida del software y explicar sus características en diferentes modelos.
- Al finalizar la unidad, el estudiante será capaz de comparar y contrastar metodologías tradicionales y ágiles, destacando sus ventajas y desventajas básicas.
- Al finalizar la unidad, el estudiante será capaz de organizar la información sobre metodologías y ciclo de vida del software para comunicarla efectivamente en presentaciones o informes.

Contenidos Temáticos

1. Introducción al Desarrollo de Software

- **Concepto de software:** Definición, tipos y características fundamentales.
- **Importancia del software en la sociedad y la industria:** Impacto y aplicaciones actuales.
- **Definición de desarrollo de software:** Procesos involucrados y objetivos principales.

2. Conceptos Fundamentales del Desarrollo de Software

- **Requisitos de software:** Qué son, tipos y su papel en el desarrollo.
- **Diseño y arquitectura de software:** Principios básicos y modelos comunes.
- **Implementación y codificación:** Prácticas estándar y calidad del código.
- **Pruebas y verificación:** Tipos de pruebas y su importancia para la calidad.
- **Mantenimiento de software:** Clasificación y retos asociados.

3. Importancia y Propósito de las Metodologías de Desarrollo de Software

- **Definición de metodología de desarrollo:** Concepto y componentes principales.
- **Objetivos de las metodologías:** Gestión, calidad, tiempos, costos y comunicación.
- **Impacto de las metodologías en proyectos reales:** Ejemplos prácticos y casos de estudio.
- **Relación entre metodología y éxito en proyectos:** Factores determinantes y mejores prácticas.

4. Ciclo de Vida del Software

- **Concepto de ciclo de vida:** Qué es y por qué es importante.
- **Fases principales del ciclo de vida:**
 - Análisis de requisitos
 - Diseño
 - Implementación
 - Pruebas
 - Despliegue
 - Mantenimiento
- **Modelos de ciclo de vida:**
 - Modelo en cascada: características y usos
 - Modelo incremental
 - Modelo en espiral
 - Modelos iterativos
- **Ventajas y limitaciones de cada modelo:** Análisis comparativo.

5. Metodologías Tradicionales vs Metodologías Ágiles

- **Metodologías tradicionales:** Características principales, ejemplos (Waterfall, RUP).
- **Metodologías ágiles:** Principios, ejemplos (Scrum, Kanban, XP).
- **Comparación detallada:** Enfoque, flexibilidad, gestión de cambios, documentación, participación del cliente.
- **Ventajas y desventajas básicas:** Contextos ideales para cada tipo de metodología.

6. Comunicación y Organización de la Información sobre Metodologías y Ciclo de Vida

- **Estructuración de la información:** Cómo organizar conceptos y datos para presentaciones o informes.
- **Herramientas y técnicas para la comunicación efectiva:** Mapas conceptuales, diagramas de flujo, tablas comparativas.
- **Buenas prácticas para presentaciones orales y escritas:** Claridad, precisión, uso de ejemplos.

Actividades

Actividad 1: Definiendo conceptos fundamentales del desarrollo de software

Objetivo: Definir los conceptos fundamentales del desarrollo de software con precisión en un contexto académico.

Descripción:

- El docente presenta brevemente los conceptos clave.
- Los estudiantes, en parejas, investigan definiciones formales y ejemplos de términos como software, ciclo de vida, requisitos, etc.
- Cada pareja prepara un glosario de términos con definiciones propias y ejemplos.
- Se realiza una puesta en común para aclarar dudas y consolidar definiciones.

Organización: Parejas

Producto esperado: Glosario de conceptos fundamentales con definiciones claras y ejemplos.

Duración estimada: 1 hora

Actividad 2: Análisis de casos reales y el impacto de las metodologías

Objetivo: Describir la importancia y el propósito de las metodologías de desarrollo de software, ejemplificando su impacto en proyectos reales.

Descripción:

- El docente presenta dos estudios de caso breves: uno donde se aplicó una metodología formal y otro sin metodología clara.
- En grupos pequeños, los estudiantes analizan las diferencias en resultados, tiempos y calidad.
- Discuten cómo la metodología influyó en el éxito o fracaso del proyecto.
- Preparan una breve presentación para compartir sus conclusiones con la clase.

Organización: Grupos de 3-4 estudiantes

Producto esperado: Presentación grupal que ejemplifique el impacto de las metodologías en proyectos reales.

Duración estimada: 1.5 horas

Actividad 3: Mapeo del ciclo de vida del software en diferentes modelos

Objetivo: Identificar las fases principales del ciclo de vida del software y explicar sus características en diferentes modelos.

Descripción:

- Se asigna a cada grupo un modelo de ciclo de vida (cascada, incremental, espiral, iterativo).
- Los grupos investigan las fases, características, ventajas y desventajas del modelo asignado.
- Construyen un diagrama visual que represente el ciclo de vida y sus fases.
- Exponen su trabajo al resto de la clase, respondiendo preguntas.

Organización: Grupos de 3-4 estudiantes

Producto esperado: Diagrama visual y explicación oral del modelo asignado.

Duración estimada: 2 horas

Actividad 4: Debate comparativo entre metodologías tradicionales y ágiles

Objetivo: Comparar y contrastar metodologías tradicionales y ágiles, destacando sus ventajas y desventajas básicas.

Descripción:

- El docente divide la clase en dos grupos: uno defiende las metodologías tradicionales y otro las ágiles.
- Cada grupo prepara argumentos basados en características, ventajas y desventajas.
- Se realiza un debate estructurado donde cada grupo expone sus puntos y refuta al otro.
- Finalmente, se realiza una reflexión grupal para identificar contextos ideales para cada enfoque.

Organización: Grupos grandes (mitad clase)

Producto esperado: Argumentos escritos y participación activa en debate.

Duración estimada: 1.5 horas

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre conceptos básicos de desarrollo de software y metodologías.

Cómo se evalúa: Cuestionario breve con preguntas abiertas y de opción múltiple sobre definiciones y conceptos generales.

Instrumento sugerido: Test escrito o digital de 10 preguntas.

Evaluación formativa

Qué se evalúa: Progreso en la comprensión de conceptos, capacidad de análisis y comunicación durante actividades.

- Revisión de glosarios creados en la actividad 1.
- Observación y retroalimentación durante las presentaciones de casos reales y diagramas de ciclo de vida.
- Evaluación de participación y calidad de argumentos en el debate.

Instrumento sugerido: Lista de cotejo, rúbrica de presentación y observación directa.

Evaluación sumativa

Qué se evalúa: Capacidad para definir conceptos, describir la importancia de las metodologías, identificar fases del ciclo de vida, comparar metodologías y comunicar información de forma organizada.

Cómo se evalúa: Informe escrito individual o en parejas que incluya:

- Definiciones precisas de conceptos fundamentales.
- Descripción y ejemplos del impacto de metodologías.
- Comparación de modelos de ciclo de vida y metodologías tradicionales vs ágiles.
- Organización clara de la información con soporte visual (diagramas, tablas).

Instrumento sugerido: Rúbrica detallada para evaluar contenido, claridad, organización y uso de ejemplos.

Unidad 2: Modelos Tradicionales de Desarrollo de Software

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de describir las características principales del Modelo en Cascada, Modelo en V y Prototipado, identificando sus fases y procesos clave.
- Al finalizar la unidad, el estudiante será capaz de comparar y contrastar los modelos tradicionales de desarrollo de software, explicando sus ventajas y desventajas en diferentes contextos de proyecto.
- Al finalizar la unidad, el estudiante será capaz de aplicar criterios para seleccionar el modelo tradicional más adecuado según los requisitos y restricciones de un caso de estudio específico.
- Al finalizar la unidad, el estudiante será capaz de analizar ejemplos prácticos de aplicación de los modelos en Cascada, en V y Prototipado, evaluando su efectividad y posibles mejoras.

Contenidos Temáticos

1. Introducción a los Modelos Tradicionales de Desarrollo de Software

- Definición y contexto histórico de los modelos tradicionales
- Importancia y relevancia en la ingeniería de software actual
- Comparación general con metodologías ágiles y modernas

2. Modelo en Cascada

- Descripción general del modelo
- Fases y procesos clave:
 - Requisitos
 - Análisis
 - Diseño
 - Implementación
 - Pruebas

- Despliegue
- Mantenimiento
- Características principales y flujo secuencial
- Ventajas y desventajas
- Contextos y tipos de proyectos adecuados para su aplicación

3. Modelo en V

- Descripción y evolución desde el modelo en cascada
- Fases y procesos clave:
 - Definición de requisitos
 - Diseño del sistema y diseño detallado
 - Codificación
 - Pruebas unitarias
 - Pruebas de integración
 - Pruebas de sistema
 - Pruebas de aceptación
- Características principales y relación entre diseño y pruebas
- Ventajas y desventajas
- Aplicaciones típicas y casos de uso

4. Modelo de Prototipado

- Concepto y finalidad del prototipado
- Tipos de prototipos:
 - Prototipo desechable
 - Prototipo evolutivo
- Fases y procesos clave:
 - Identificación de requisitos preliminares
 - Construcción del prototipo
 - Evaluación y retroalimentación
 - Refinamiento o construcción del sistema final
- Características principales y beneficios
- Ventajas y desventajas
- Contextos en los que es adecuado utilizar prototipos

5. Comparación y Contraste de Modelos Tradicionales

- Análisis comparativo de fases y procesos
- Ventajas y limitaciones relativas entre cascada, V y prototipado
- Criterios para seleccionar el modelo adecuado
 - Requisitos del proyecto
 - Complejidad y tamaño
 - Flexibilidad y cambios esperados
 - Recursos y tiempo disponible

6. Análisis de Casos Prácticos

- Ejemplos reales y simulados de aplicación del modelo en cascada
- Ejemplos prácticos aplicando el modelo en V
- Estudios de caso con prototipado y evaluación de resultados
- Identificación de mejoras y recomendaciones para cada caso

Actividades

Actividad 1: Mapa conceptual de Modelos Tradicionales

Objetivo: Describir las características principales del Modelo en Cascada, Modelo en V y Prototipado, identificando sus fases y procesos clave.

Descripción:

- Los estudiantes investigan individualmente las fases y características de cada modelo.
- En parejas, elaboran un mapa conceptual integrando los modelos, señalando sus fases y procesos clave.
- Discuten y refinan el mapa en grupo para asegurar precisión y claridad.
- Presentan el mapa conceptual a la clase y responden preguntas.

Organización: Individual para investigación, luego parejas y grupos.

Producto esperado: Mapa conceptual digital o impreso que muestra la estructura y fases de los modelos tradicionales.

Duración estimada: 2 horas.

Actividad 2: Debate Comparativo de Modelos

Objetivo: Comparar y contrastar los modelos tradicionales, explicando sus ventajas y desventajas en diferentes contextos de proyecto.

Descripción:

- Dividir la clase en tres grupos, cada uno defendiendo uno de los modelos (Cascada, V, Prototipado).
- Cada grupo prepara argumentos basados en ventajas, desventajas y escenarios adecuados.
- Se realiza un debate estructurado donde cada grupo expone y responde a preguntas de los demás y del docente.

- Al finalizar, se hace una reflexión conjunta sobre los puntos clave y diferencias.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Argumentos escritos y participación en debate.

Duración estimada: 2 horas.

Actividad 3: Selección de Modelo para Caso de Estudio

Objetivo: Aplicar criterios para seleccionar el modelo tradicional más adecuado según los requisitos y restricciones de un caso de estudio específico.

Descripción:

- Se presenta un caso de estudio detallado con requisitos y restricciones.
- En grupos, los estudiantes analizan el caso, identifican necesidades clave y seleccionan el modelo tradicional que consideran más adecuado.
- Justifican su elección con base en criterios aprendidos y elaboran un informe breve.
- Exponen su propuesta y discuten con otros grupos.

Organización: Grupos de 3-4 estudiantes.

Producto esperado: Informe escrito y presentación oral.

Duración estimada: 3 horas.

Actividad 4: Análisis Crítico de Ejemplos Prácticos

Objetivo: Analizar ejemplos prácticos de aplicación de los modelos en Cascada, en V y Prototipado, evaluando su efectividad y posibles mejoras.

Descripción:

- Se proporcionan estudios de caso reales o simulados para cada modelo.
- Individualmente, los estudiantes analizan el caso, identifican los puntos fuertes y débiles de la aplicación del modelo.
- En parejas, discuten posibles mejoras y alternativas.
- El docente modera una sesión plenaria para compartir conclusiones y aprendizajes.

Organización: Individual y luego parejas.

Producto esperado: Informe de análisis crítico y síntesis de mejoras propuestas.

Duración estimada: 2 horas.

Evaluación

Evaluación Diagnóstica

Qué se evalúa: Conocimientos previos sobre modelos de desarrollo de software y comprensión inicial de sus características.

Cómo se evalúa: Cuestionario escrito con preguntas de opción múltiple y de respuesta corta sobre conceptos básicos y fases de modelos tradicionales.

Instrumento sugerido: Cuestionario en línea o impreso de 10-15 preguntas.

Evaluación Formativa

Qué se evalúa: Progreso en la comprensión y aplicación de los modelos; capacidad para comparar modelos y seleccionar el más adecuado.

Cómo se evalúa: Revisión de productos de actividades (mapa conceptual, argumentos del debate, informes de selección de modelo, análisis crítico), retroalimentación continua durante discusiones y actividades.

Instrumento sugerido: Rúbricas para evaluación de mapas conceptuales, debates, informes escritos y presentaciones orales; observación y notas del docente.

Evaluación Sumativa

Qué se evalúa: Dominio integral de los modelos tradicionales, capacidad analítica y crítica, aplicación práctica en casos reales o simulados.

Cómo se evalúa: Examen escrito que incluye preguntas teóricas y análisis de casos; proyecto final donde se debe seleccionar y justificar un modelo para un caso complejo, y analizar su aplicación.

Instrumento sugerido: Examen escrito tipo ensayo y/o análisis de caso, entrega de proyecto final con rúbrica detallada.

Unidad 3: Modelos Iterativos e Incrementales

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de explicar los conceptos fundamentales y características del modelo iterativo e incremental, incluyendo el modelo Espiral y RUP, identificando sus ventajas y desventajas en diferentes contextos de desarrollo.
- Al finalizar la unidad, el estudiante será capaz de comparar y contrastar el modelo Espiral y el proceso RUP, evaluando sus aplicaciones prácticas y su impacto en la gestión y calidad de proyectos de software.
- Al finalizar la unidad, el estudiante será capaz de aplicar principios del modelo iterativo e incremental para diseñar un plan de desarrollo progresivo, adaptando las prácticas a escenarios específicos de proyectos de software.
- Al finalizar la unidad, el estudiante será capaz de analizar casos de estudio que empleen modelos iterativos e incrementales, identificando áreas de mejora y proponiendo ajustes para optimizar el proceso de desarrollo.

Contenidos Temáticos

1. Introducción a los Modelos Iterativos e Incrementales

- Concepto y definición del desarrollo iterativo e incremental: explicación de cómo estos modelos permiten la construcción progresiva del software mediante ciclos repetidos y entregas parciales.
- Comparación con modelos tradicionales (cascada): diferencias en enfoque, flexibilidad y gestión de riesgos.
- Beneficios generales de los modelos iterativos e incrementales: adaptación al cambio, mejora continua y reducción de riesgos.

2. Modelo Iterativo e Incremental: Fundamentos y Características

- Descripción detallada del ciclo iterativo: planificación, diseño, construcción, evaluación y revisión.
- Incrementos funcionales: definición y ejemplos de entregas parciales de software con valor agregado.
- Gestión del alcance y cambios durante el proceso iterativo.
- Ventajas y desventajas de este enfoque en contextos reales de desarrollo.

3. Modelo Espiral

- Origen y contexto histórico del Modelo Espiral.
- Estructura del modelo: fases del ciclo espiral (planeación, análisis de riesgos, ingeniería, evaluación del cliente).
- La importancia de la gestión de riesgos como eje central del modelo.
- Ventajas y limitaciones del Modelo Espiral.
- Casos de aplicación típicos y escenarios recomendados.

4. Rational Unified Process (RUP)

- Introducción a RUP: definición y propósito.
- Fases de RUP: inicio, elaboración, construcción y transición.
- Roles, actividades y artefactos en RUP.
- Principios iterativos y gestión de riesgos en RUP.
- Ventajas y desventajas del proceso RUP.
- Ejemplos prácticos de implementación de RUP en proyectos reales.

5. Comparación entre el Modelo Espiral y RUP

- Análisis comparativo: estructura, enfoque en riesgos, flexibilidad y documentación.
- Impacto en la gestión de proyectos y calidad del software.
- Contextos y tipos de proyectos donde cada modelo es más adecuado.
- Lecciones aprendidas y mejores prácticas derivadas de ambos modelos.

6. Diseño de un Plan de Desarrollo Iterativo e Incremental

- Principios clave para el diseño de planes de desarrollo progresivo.
- Identificación y priorización de requisitos para iteraciones e incrementos.
- Definición de entregables y criterios de aceptación en cada incremento.

- Adaptación del plan a diferentes escenarios de proyectos (tamaño, complejidad, equipo).
- Herramientas y técnicas para la planificación iterativa.

7. Análisis de Casos de Estudio con Modelos Iterativos e Incrementales

- Presentación de casos reales que utilizan Modelo Espiral y RUP.
- Identificación de fortalezas y debilidades en la aplicación de los modelos.
- Propuesta de mejoras y ajustes para optimizar el proceso de desarrollo.
- Discusión sobre el impacto de las decisiones en la calidad y gestión del proyecto.

Actividades

Actividad 1: Debate Comparativo sobre Modelos Iterativos e Incrementales

Objetivo: Contribuye al primer y segundo objetivo de la unidad, permitiendo a los estudiantes explicar y comparar los modelos Espiral y RUP.

Descripción:

- Dividir la clase en dos grupos: uno defenderá el modelo Espiral y el otro el proceso RUP.
- Cada grupo investigará las características, ventajas y desventajas de su modelo asignado.
- Prepararán una presentación breve para argumentar por qué su modelo es adecuado en ciertos contextos.
- Se realizará un debate donde ambos grupos expongan sus argumentos y respondan preguntas del otro grupo y del docente.
- Finalmente, se hará una reflexión conjunta sobre las diferencias y aplicaciones prácticas de ambos modelos.

Organización: Grupos

Producto esperado: Presentación y debate con argumentos fundamentados.

Duración estimada: 90 minutos

Actividad 2: Taller de Diseño de Plan Iterativo e Incremental

Objetivo: Aplica al tercer objetivo, diseñar un plan de desarrollo progresivo adaptado a un escenario específico.

Descripción:

- Presentar un escenario práctico de proyecto de software (por ejemplo, desarrollo de una aplicación móvil).
- Los estudiantes, en grupos pequeños, identificarán requisitos prioritarios y definirán iteraciones e incrementos.
- Elaborarán un plan que incluya entregables, actividades y criterios de aceptación para cada iteración.
- Presentarán el plan al grupo y recibirán retroalimentación para ajustes.

Organización: Grupos

Producto esperado: Plan de desarrollo iterativo e incremental documentado.

Duración estimada: 120 minutos

Actividad 3: Análisis de Caso de Estudio y Propuesta de Mejoras

Objetivo: Relacionado con el cuarto objetivo, analizar casos y proponer ajustes para optimizar el desarrollo.

Descripción:

- Proporcionar un caso de estudio real o simulado donde se haya aplicado un modelo iterativo (Espiral o RUP).
- Individualmente o en parejas, los estudiantes analizarán el caso, identificando problemas, fortalezas y debilidades.
- Elaborarán un informe con propuestas de mejora para optimizar el proceso de desarrollo.
- Se realizará una sesión de discusión grupal para compartir análisis y propuestas.

Organización: Individual o parejas

Producto esperado: Informe de análisis y propuestas de mejora.

Duración estimada: 90 minutos

Actividad 4: Quiz Interactivo sobre Conceptos Clave

Objetivo: Refuerzo y evaluación formativa de los conceptos fundamentales del modelo iterativo e incremental, Espiral y RUP.

Descripción:

- Realizar un quiz en línea o presencial con preguntas de opción múltiple, verdadero/falso y de respuesta corta.
- Las preguntas cubrirán definiciones, características, ventajas/desventajas y comparaciones entre los modelos.
- Los resultados se discutirán brevemente para aclarar dudas y reforzar aprendizajes.

Organización: Individual

Producto esperado: Resultados del quiz y retroalimentación.

Duración estimada: 30 minutos

Evaluación

Evaluación Diagnóstica

Qué se evalúa: Conocimientos previos sobre modelos de desarrollo de software, especialmente modelos iterativos e incrementales.

Cómo se evalúa: Cuestionario corto con preguntas básicas sobre modelos de desarrollo tradicionales y sus limitaciones.

Instrumento sugerido: Prueba escrita o quiz en plataforma digital con preguntas de opción múltiple y respuesta corta.

Evaluación Formativa

Qué se evalúa: Comprensión continua de conceptos y habilidades para aplicar los modelos en diseño y análisis.

Cómo se evalúa: Observación y revisión de actividades prácticas: debate, taller de diseño y análisis de casos.

Instrumento sugerido: Rúbricas para evaluar participación, argumentación, calidad del plan de desarrollo y profundidad del análisis de casos.

Evaluación Sumativa

Qué se evalúa: Dominio integral de los conceptos, capacidad de comparación crítica, aplicación práctica y análisis de casos.

Cómo se evalúa: Examen escrito o proyecto final que incluya:

- Preguntas teóricas sobre modelos iterativos, Espiral y RUP.
- Ejercicio de comparación y contraste entre modelos.
- Diseño de un plan de desarrollo iterativo para un caso dado.
- Análisis crítico de un caso de estudio con propuestas de mejora.

Instrumento sugerido: Examen escrito con combinación de preguntas teóricas y prácticas o entrega de proyecto final con rúbrica detallada.

Unidad 4: Fundamentos de las Metodologías Ágiles

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar y explicar los principios fundamentales del Manifiesto Ágil, evaluando su impacto en el desarrollo de software.
- Al finalizar la unidad, el estudiante será capaz de analizar los valores centrales de las metodologías ágiles y comparar su aplicación con metodologías tradicionales en proyectos de software.
- Al finalizar la unidad, el estudiante será capaz de describir las características distintivas de las principales metodologías ágiles, ejemplificando su uso en contextos reales de desarrollo.
- Al finalizar la unidad, el estudiante será capaz de interpretar situaciones de proyectos de software para justificar la elección de enfoques ágiles basados en sus principios y valores.

Contenidos Temáticos

1. Introducción a las Metodologías Ágiles

- **Contexto histórico y evolución:** Se abordará la transición desde metodologías tradicionales hacia enfoques ágiles, destacando los desafíos del desarrollo de software que motivaron este cambio.
- **Definición y características generales:** Explicación de qué son las metodologías ágiles, sus principios básicos y características distintivas frente a metodologías tradicionales.

2. El Manifiesto Ágil y sus Principios Fundamentales

- **Origen y propósito del Manifiesto Ágil:** Contextualización de la creación del manifiesto en 2001 por un grupo de expertos y su relevancia actual.

- **Los cuatro valores del Manifiesto Ágil:**

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Se explicará el significado y la implicancia práctica de cada valor.

- **Los doce principios ágiles:** Desglose y explicación detallada de cada principio, con ejemplos de aplicación en proyectos de software.
- **Impacto de los principios en el desarrollo de software:** Análisis de cómo estos principios mejoran la calidad, flexibilidad y satisfacción del cliente.

3. Valores Ágiles vs. Metodologías Tradicionales

- **Comparación de valores y enfoques:** Contraste entre metodologías tradicionales (como el modelo en cascada) y ágiles en cuanto a planificación, documentación, gestión de cambios y comunicación.
- **Ventajas y limitaciones de cada enfoque:** Discusión crítica sobre cuándo es más apropiado usar metodologías ágiles o tradicionales.

4. Principales Metodologías Ágiles

- **Scrum:** Estructura, roles (Scrum Master, Product Owner, equipo), eventos (sprints, reuniones diarias), artefactos (backlog, incrementos).
- **Kanban:** Conceptos clave, visualización del flujo de trabajo, gestión del trabajo en progreso, mejora continua.
- **Extreme Programming (XP):** Prácticas esenciales, desarrollo incremental, enfoque en calidad y pruebas.
- **Lean Software Development:** Principios derivados de Lean Manufacturing aplicados al software.
- **Comparación y selección de metodologías:** Ejemplos de escenarios reales y criterios para elegir la metodología ágil más adecuada.

5. Aplicación Práctica de los Principios y Valores Ágiles en Proyectos de Software

- **Interpretación de casos y situaciones reales:** Análisis de contextos de proyectos para identificar la conveniencia de aplicar enfoques ágiles.
- **Justificación de la elección de metodologías ágiles:** Argumentación basada en los principios y valores para seleccionar un enfoque específico.
- **Impacto esperado en resultados y equipo:** Cómo la adopción de metodologías ágiles influye en la productividad, calidad y satisfacción del cliente.

Actividades

Actividad 1: Análisis y exposición del Manifiesto Ágil

Objetivo: Identificar y explicar los principios fundamentales del Manifiesto Ágil y evaluar su impacto en el desarrollo de software.

Descripción paso a paso:

- El docente divide a los estudiantes en grupos pequeños (3-4 integrantes).
- A cada grupo se le asigna uno o dos valores y varios principios del Manifiesto Ágil para analizar en profundidad.
- Los grupos investigan el significado, implicancias prácticas y ejemplos reales de los valores y principios asignados.
- Preparan una presentación corta (10 minutos) para exponer sus hallazgos al resto de la clase.
- Se realiza una discusión colectiva sobre el impacto de estos valores y principios en proyectos actuales.

Organización: Grupos

Producto esperado: Presentación grupal con análisis y ejemplos prácticos.

Duración estimada: 90 minutos

Actividad 2: Comparación práctica entre metodologías tradicionales y ágiles

Objetivo: Analizar los valores centrales de las metodologías ágiles y comparar su aplicación con metodologías tradicionales en proyectos de software.

Descripción paso a paso:

- Se presenta a los estudiantes un caso hipotético de proyecto de desarrollo de software con requisitos cambiantes.
- Individualmente, cada estudiante elabora una tabla comparativa que contraste el enfoque tradicional y el ágil en aspectos como planificación, gestión de cambios, documentación y comunicación.
- En parejas, discuten sus comparaciones y generan una conclusión sobre cuál enfoque es más adecuado para el caso.
- Compartir conclusiones en plenaria para consolidar aprendizajes.

Organización: Individual y parejas

Producto esperado: Tabla comparativa y conclusión escrita.

Duración estimada: 60 minutos

Actividad 3: Simulación de un Sprint Scrum

Objetivo: Describir las características distintivas de las principales metodologías ágiles, ejemplificando su uso en contextos reales de desarrollo.

Descripción paso a paso:

- Se divide a los estudiantes en equipos de 5 personas, asignando roles de Scrum Master, Product Owner y equipo de desarrollo.
- Se les entrega un proyecto simple (por ejemplo, desarrollo de una app básica) con un backlog inicial.
- Realizan una planificación de sprint, selección de tareas y ejecución de reuniones diarias simuladas (10 minutos cada una).

- Al finalizar el sprint simulado, presentan el incremento de producto y reflexionan sobre la experiencia de aplicar Scrum.

Organización: Grupos

Producto esperado: Reporte de sprint y presentación del incremento simulado.

Duración estimada: 120 minutos

Actividad 4: Estudio de caso para elección de metodología ágil

Objetivo: Interpretar situaciones de proyectos de software para justificar la elección de enfoques ágiles basados en sus principios y valores.

Descripción paso a paso:

- Se presentan varios escenarios distintos de proyectos con variables como tamaño del equipo, estabilidad de requisitos, tiempo disponible y experiencia previa.
- Los estudiantes, en grupos pequeños, analizan cada escenario y seleccionan la metodología ágil más adecuada, justificando su elección con base en los principios y valores estudiados.
- Preparan un informe breve y exponen sus decisiones al resto de la clase.

Organización: Grupos

Producto esperado: Informe escrito y presentación oral.

Duración estimada: 90 minutos

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre metodologías de desarrollo de software y comprensión inicial de conceptos ágiles.

Cómo se evalúa: Preguntas abiertas y de opción múltiple sobre conceptos básicos.

Instrumento sugerido: Cuestionario breve en línea o en papel al inicio de la unidad.

Evaluación formativa

Qué se evalúa: Participación y comprensión progresiva de los valores, principios y metodologías ágiles durante las actividades.

Cómo se evalúa: Observación y retroalimentación durante actividades grupales, revisión de productos parciales (tablas comparativas, reportes de sprint, análisis de casos).

Instrumento sugerido: Rúbricas para presentaciones y productos escritos, listas de cotejo para participación y aportes en discusiones.

Evaluación sumativa

Qué se evalúa: Capacidad para identificar, explicar y aplicar principios y valores ágiles, así como justificar la elección de metodologías en contextos reales.

Cómo se evalúa: Examen escrito con preguntas de desarrollo y análisis de casos prácticos; entrega de un trabajo integrador final donde se aplique un enfoque ágil a un proyecto propuesto.

Instrumento sugerido: Examen final y rubrica de evaluación para trabajo integrador.

Unidad 5: Scrum: Roles, Eventos y Artefactos

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar y describir los roles principales en Scrum, incluyendo Scrum Master, Product Owner y el equipo de desarrollo, diferenciando sus responsabilidades dentro del proceso.
- Al finalizar la unidad, el estudiante será capaz de analizar y explicar los eventos clave de Scrum, tales como Sprint, Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective, estableciendo su propósito y dinámica en proyectos de software.
- Al finalizar la unidad, el estudiante será capaz de clasificar y evaluar los artefactos de Scrum, como Product Backlog, Sprint Backlog e Incremento, relacionándolos con la gestión y seguimiento del proyecto.
- Al finalizar la unidad, el estudiante será capaz de aplicar los conceptos de roles, eventos y artefactos de Scrum en un caso práctico simulado, demostrando comprensión integral de la metodología en el contexto de proyectos de software.

Contenidos Temáticos

1. Introducción a Scrum

- Definición y origen de Scrum: Se abordará el contexto histórico y la necesidad que llevó a la creación de Scrum como metodología ágil para desarrollo de software.
- Principios y valores fundamentales de Scrum: Descripción de los valores (compromiso, coraje, enfoque, apertura y respeto) y principios que guían la práctica de Scrum.
- Visión general del marco de trabajo Scrum: Presentación de la estructura básica de Scrum, sus roles, eventos y artefactos.

2. Roles en Scrum

- Scrum Master: Responsabilidades y funciones principales, facilitación del equipo, eliminación de impedimentos, aseguramiento de la correcta aplicación de Scrum.
- Product Owner: Gestión del Product Backlog, priorización de requisitos, representación de los intereses del cliente y stakeholders.
- Equipo de desarrollo: Composición, auto-organización, responsabilidad en la entrega del incremento de producto potencialmente entregable.

- Diferenciación y colaboración entre roles: Cómo interactúan y complementan sus funciones los roles dentro del equipo Scrum.

3. Eventos clave en Scrum

- Sprint: Concepto, duración típica, objetivo y definición de "Incremento".
- Sprint Planning: Objetivos, actividades, definición del Sprint Backlog y compromiso del equipo.
- Daily Scrum: Propósito, estructura, duración, y beneficios para la sincronización del equipo.
- Sprint Review: Revisión del trabajo realizado, presentación del incremento, obtención de feedback de stakeholders.
- Sprint Retrospective: Reflexión sobre el proceso, identificación de mejoras continuas para el siguiente Sprint.

4. Artefactos en Scrum

- Product Backlog: Definición, características, gestión y mantenimiento, priorización y refinamiento.
- Sprint Backlog: Componentes, selección de ítems del Product Backlog para el Sprint, tareas y plan de trabajo del equipo.
- Incremento: Definición del incremento de producto, criterios de "Done" (terminado), importancia para la entrega continua de valor.
- Transparencia y visibilidad de los artefactos: Cómo los artefactos facilitan la comunicación y seguimiento del proyecto.

5. Aplicación práctica de Scrum

- Simulación de un proyecto Scrum: Asignación de roles, planificación y ejecución de un Sprint.
- Desarrollo de un Product Backlog inicial y planificación del Sprint Backlog.
- Conducción de eventos Scrum en la simulación: Daily Scrum, Sprint Review y Sprint Retrospective.
- Evaluación y reflexión sobre la experiencia de trabajo con Scrum: fortalezas, desafíos y aprendizajes.

Actividades

Actividad 1: Identificación y análisis de roles en Scrum

Objetivo: Al finalizar, el estudiante podrá identificar y describir los roles principales en Scrum y sus responsabilidades.

Descripción:

- El docente divide a los estudiantes en grupos pequeños (3-4 personas).
- Cada grupo recibe un caso práctico donde deben asignar roles Scrum a diferentes perfiles y justificar las responsabilidades de cada uno.
- Los grupos presentan sus asignaciones y se discuten en plenaria para aclarar dudas y profundizar en las funciones de cada rol.

Organización: Grupos

Producto esperado: Presentación grupal con roles asignados y justificación.

Duración estimada: 60 minutos

Actividad 2: Simulación de eventos Scrum en un Sprint

Objetivo: Analizar y explicar los eventos clave de Scrum estableciendo su propósito y dinámica.

Descripción:

- El grupo simula un proyecto software, asignando roles Scrum.
- Ejecutan un ciclo completo de Sprint, realizando Sprint Planning, Daily Scrum (simulando varios días), Sprint Review y Sprint Retrospective.
- Durante la simulación, cada estudiante toma notas sobre el propósito y resultados de cada evento.
- Posteriormente, elaboran un reporte breve explicando cada evento, su dinámica y aprendizajes obtenidos.

Organización: Grupos

Producto esperado: Reporte escrito sobre los eventos Scrum y su función en la simulación.

Duración estimada: 3 horas (puede dividirse en sesiones)

Actividad 3: Taller de clasificación y evaluación de artefactos Scrum

Objetivo: Clasificar y evaluar los artefactos de Scrum relacionándolos con la gestión y seguimiento del proyecto.

Descripción:

- Los estudiantes reciben varios ejemplos de Product Backlogs, Sprint Backlogs e Incrementos (documentos o descripciones).
- Individualmente o en parejas, clasifican cada ejemplo en el artefacto correspondiente y explican cómo contribuye al proyecto.
- Discusión grupal para comparar respuestas y clarificar aspectos de cada artefacto.

Organización: Individual o parejas

Producto esperado: Documento con clasificación y explicación de artefactos.

Duración estimada: 90 minutos

Actividad 4: Caso práctico integral de Scrum

Objetivo: Aplicar los conceptos de roles, eventos y artefactos de Scrum en un caso práctico simulado.

Descripción:

- Se presenta un caso complejo para ser desarrollado en grupos, donde deben diseñar todo el proceso Scrum: definir roles, planificar eventos, y gestionar artefactos.
- Los estudiantes elaboran un plan de proyecto Scrum, ejecutan un Sprint simulado y documentan el proceso.
- Presentan sus resultados, reflexiones y propuestas de mejora.

Organización: Grupos

Producto esperado: Informe integral y presentación del proyecto Scrum simulado.

Duración estimada: 4 horas (puede repartirse en varias sesiones)

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre metodologías ágiles y conceptos básicos de Scrum.

Cómo se evalúa: Cuestionario de opción múltiple y preguntas abiertas breves.

Instrumento sugerido: Test en línea o papel con 10 preguntas sobre fundamentos de Scrum.

Evaluación formativa

Qué se evalúa: Participación en actividades prácticas, comprensión de roles, eventos y artefactos.

Cómo se evalúa: Observación directa, revisión de productos parciales (presentaciones, reportes, documentos), retroalimentación continua.

Instrumento sugerido: Rúbrica para evaluar presentaciones grupales, reportes escritos y participación en discusión.

Evaluación sumativa

Qué se evalúa: Integración y aplicación de conocimientos en un caso práctico integral de Scrum.

Cómo se evalúa: Informe final y presentación del caso práctico, demostrando comprensión de roles, eventos y artefactos.

Instrumento sugerido: Rúbrica detallada que considere precisión conceptual, aplicación práctica, trabajo en equipo y calidad de la presentación.

Unidad 6: Kanban y Otras Metodologías Ágiles

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de describir los principios y prácticas fundamentales de Kanban, identificando sus componentes clave y su impacto en la gestión del flujo de trabajo.
- Al finalizar la unidad, el estudiante será capaz de comparar Kanban con otras metodologías ágiles complementarias, evaluando sus ventajas y limitaciones en contextos específicos de desarrollo de software.
- Al finalizar la unidad, el estudiante será capaz de aplicar técnicas visuales de gestión y control de tareas mediante tableros Kanban, adaptándolas a escenarios reales de proyectos de software.
- Al finalizar la unidad, el estudiante será capaz de analizar y proponer mejoras en procesos de desarrollo utilizando metodologías ágiles, basándose en métricas de rendimiento y retroalimentación continua.
- Al finalizar la unidad, el estudiante será capaz de comunicar efectivamente los beneficios y desafíos de implementar Kanban y otras metodologías ágiles en presentaciones académicas o profesionales.

Contenidos Temáticos

1. Introducción a Kanban

- **Origen y evolución de Kanban:** Historia de Kanban, desde su creación en la manufactura hasta su adopción en desarrollo de software.
- **Principios fundamentales de Kanban:** Visualizar el trabajo, limitar el trabajo en progreso (WIP), gestionar el flujo, hacer políticas explícitas y mejorar colaborativamente.
- **Componentes clave de Kanban:** Tablero Kanban, tarjetas, columnas, límites WIP y políticas explícitas.
- **Impacto de Kanban en la gestión del flujo de trabajo:** Cómo Kanban mejora la visibilidad, reduce cuellos de botella y optimiza el rendimiento del equipo.

2. Otras Metodologías Ágiles Complementarias

- **Scrum:** Marco de trabajo ágil basado en iteraciones, roles definidos y eventos planificados.
- **Extreme Programming (XP):** Prácticas técnicas enfocadas en la calidad del código y la colaboración continua.
- **Lean Software Development:** Principios orientados a eliminar desperdicios y mejorar la eficiencia del proceso.
- **Comparación entre Kanban y otras metodologías ágiles:** Ventajas, limitaciones y casos de uso para cada metodología en proyectos de software.

3. Técnicas Visuales de Gestión con Tableros Kanban

- **Diseño y estructura del tablero Kanban:** Columnas, filas, tipos de tarjetas y colores para representar estados y prioridades.
- **Gestión de tareas y flujo de trabajo:** Creación, asignación, seguimiento y control de tareas usando el tablero.
- **Adaptación del tablero a escenarios reales:** Ajustes para equipos distribuidos, proyectos multi-equipo y diferentes tipos de trabajo.
- **Herramientas digitales para Kanban:** Introducción a herramientas comunes (Trello, Jira, Azure DevOps) y su integración en el proceso.

4. Análisis y Mejora Continua en Procesos Ágiles

- **Métricas clave en Kanban y metodologías ágiles:** Tiempo de ciclo, tiempo de entrega, throughput, acumulación de trabajo en progreso.
- **Análisis de datos para la mejora del proceso:** Identificación de cuellos de botella y áreas de mejora mediante métricas.
- **Retroalimentación continua:** Técnicas para recoger feedback y ajustar el proceso de desarrollo.
- **Propuesta de mejoras basadas en evidencias:** Formular estrategias para optimizar el flujo y la calidad del trabajo.

5. Comunicación de Beneficios y Desafíos de Kanban y Otras Metodologías Ágiles

- **Elementos clave para la comunicación efectiva:** Claridad, estructura, evidencia y adecuación al público.
- **Beneficios de implementar Kanban y metodologías ágiles:** Flexibilidad, visibilidad, mejora continua y alineación con objetivos de negocio.

- **Desafíos comunes y cómo abordarlos:** Resistencia al cambio, adaptación cultural, integración con procesos existentes.
- **Preparación y presentación de informes y presentaciones:** Uso de recursos visuales, storytelling y técnicas persuasivas en contextos académicos y profesionales.

Actividades

Actividad 1: Construcción y análisis de un tablero Kanban

Objetivo: Aplicar técnicas visuales de gestión y control de tareas mediante tableros Kanban (Objetivo 3).

Descripción:

- Formar grupos de 4 estudiantes.
- Asignar un proyecto de desarrollo de software simple (por ejemplo, desarrollo de una aplicación de lista de tareas).
- Cada grupo diseñará y construirá un tablero Kanban físico o digital, definiendo columnas, tarjetas y políticas.
- Simularán el flujo de trabajo moviendo tareas a través de las columnas durante una sesión controlada.
- Analizarán el flujo, identificando posibles cuellos de botella y proponiendo ajustes en el tablero.

Organización: Grupos

Producto esperado: Tablero Kanban funcional, análisis escrito de flujo y propuesta de mejoras.

Duración estimada: 2 horas

Actividad 2: Debate comparativo entre Kanban y Scrum

Objetivo: Comparar Kanban con otras metodologías ágiles complementarias (Objetivo 2).

Descripción:

- Dividir la clase en dos equipos.
- Un equipo defenderá el uso de Kanban y el otro el uso de Scrum en proyectos de software.
- Prepararán argumentos basados en ventajas, limitaciones y casos de uso.
- Realizarán un debate estructurado con presentación de argumentos y refutaciones.
- Al finalizar, cada equipo redactará un resumen de conclusiones y aprendizajes.

Organización: Grupos (equipos)

Producto esperado: Resumen escrito con conclusiones y argumentos principales.

Duración estimada: 1.5 horas

Actividad 3: Análisis de métricas y propuesta de mejora

Objetivo: Analizar y proponer mejoras en procesos de desarrollo utilizando metodologías ágiles, basándose en métricas y retroalimentación continua (Objetivo 4).

Descripción:

- Se proporcionará a los estudiantes un conjunto de datos ficticios de métricas Kanban (tiempo de ciclo, WIP, throughput).
- En parejas, analizarán la información para identificar problemas o ineficiencias en el proceso.
- Desarrollarán una propuesta de mejora fundamentada en los hallazgos y principios ágiles.
- Presentarán la propuesta en un informe breve y una presentación oral.

Organización: Parejas

Producto esperado: Informe escrito y presentación oral.

Duración estimada: 2 horas

Actividad 4: Presentación sobre beneficios y desafíos de Kanban

Objetivo: Comunicar efectivamente los beneficios y desafíos de implementar Kanban y otras metodologías ágiles (Objetivo 5).

Descripción:

- Individualmente, cada estudiante preparará una presentación de 5 minutos dirigida a un público profesional.
- La presentación debe incluir: introducción a Kanban, beneficios, desafíos comunes y recomendaciones para la implementación.
- Se fomentará el uso de recursos visuales y la claridad en la comunicación.
- Se realizará una sesión de presentaciones con retroalimentación del docente y compañeros.

Organización: Individual

Producto esperado: Presentación oral con apoyo visual.

Duración estimada: 1.5 horas (considerando todas las presentaciones)

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre metodologías ágiles y gestión de proyectos.

Cómo se evalúa: Cuestionario breve tipo test o preguntas abiertas al inicio de la unidad.

Instrumento sugerido: Test digital o en papel con 10 preguntas sobre conceptos básicos de Kanban y metodologías ágiles.

Evaluación formativa

Qué se evalúa: Participación y desempeño en actividades prácticas, comprensión de conceptos y aplicación de técnicas.

Cómo se evalúa: Observación directa durante actividades, revisión de productos entregados (tableros, informes, presentaciones), retroalimentación continua.

Instrumento sugerido: Rúbricas para evaluación de tableros Kanban, debates, informes y presentaciones orales.

Evaluación sumativa

Qué se evalúa: Comprensión integral de los principios y prácticas de Kanban y metodologías ágiles, capacidad analítica, aplicación práctica y comunicación efectiva.

Cómo se evalúa: Examen escrito con preguntas teóricas y análisis de casos, junto con una entrega final que incluya un análisis de métricas y una propuesta de mejora documentada.

Instrumento sugerido: Examen parcial y entrega de un proyecto escrito individual o grupal que integre los aprendizajes de la unidad.

Unidad 7: Herramientas para la Gestión de Proyectos Ágiles

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar las principales herramientas digitales utilizadas en la gestión de proyectos ágiles, diferenciando sus funciones y características.
- Al finalizar la unidad, el estudiante será capaz de aplicar software de seguimiento y control para planificar y monitorear proyectos ágiles en escenarios simulados.
- Al finalizar la unidad, el estudiante será capaz de analizar la integración de herramientas digitales en el proceso ágil para mejorar la colaboración y la eficiencia del equipo de desarrollo.
- Al finalizar la unidad, el estudiante será capaz de evaluar críticamente el uso de diferentes herramientas ágiles en función de los requisitos y contexto del proyecto, proponiendo recomendaciones para su selección y uso adecuado.

Contenidos Temáticos

1. Introducción a las herramientas digitales en metodologías ágiles

- Definición y propósito de las herramientas digitales en proyectos ágiles: se explicará cómo estas herramientas facilitan la planificación, seguimiento, comunicación y colaboración dentro de un equipo ágil.
- Importancia de la gestión digital en el contexto ágil: se abordará la necesidad de herramientas que soporten iteraciones rápidas, cambios frecuentes y la visibilidad del progreso.

2. Principales herramientas digitales para la gestión de proyectos ágiles

• Software de seguimiento y control

- Jira: características, funciones principales y casos de uso en Scrum y Kanban.
- Trello: estructura basada en tableros y tarjetas, simplicidad y flexibilidad para equipos pequeños.
- Asana: gestión de tareas y flujos de trabajo con enfoque en colaboración y visualización.
- Azure DevOps: integración con desarrollo, repositorios y pipelines CI/CD.

• Herramientas para comunicación y colaboración

- Slack: canales, integración con otras herramientas y notificaciones en tiempo real.

- Microsoft Teams: reuniones, chats y colaboración integrada con Office 365.
- Confluence: documentación colaborativa y gestión del conocimiento.

- **Herramientas para planificación ágil**

- VersionOne: planificación, seguimiento y reporte de proyectos ágiles.
- Targetprocess: visualización y gestión de portafolios ágiles.

3. Funcionalidades y características clave de las herramientas ágiles

- Gestión de backlog: creación, priorización y mantenimiento.
- Tableros Kanban y Scrum: diseño, uso y personalización.
- Reportes y métricas ágiles: burndown charts, velocity, lead time.
- Integraciones con repositorios y herramientas de CI/CD.
- Soporte para colaboración en tiempo real y comunicación integrada.

4. Aplicación práctica de software de seguimiento y control en proyectos ágiles

- Configuración inicial de un proyecto ágil en Jira y Trello.
- Creación y gestión de user stories, tareas y sprints.
- Monitoreo del progreso usando tableros y reportes.
- Simulación de un ciclo completo de sprint con seguimiento en el software.

5. Integración de herramientas digitales en el proceso ágil para mejorar la colaboración

- Sincronización entre herramientas de gestión y comunicación.
- Automatización de notificaciones y alertas.
- Fomento de la transparencia y visibilidad del trabajo en equipo.
- Casos de estudio sobre mejora de eficiencia mediante integración de herramientas.

6. Evaluación crítica y selección de herramientas ágiles según requisitos y contexto

- Criterios para seleccionar herramientas: tamaño de equipo, tipo de proyecto, presupuesto, curva de aprendizaje.
- Ventajas y limitaciones de las herramientas más populares.
- Adaptabilidad a diferentes metodologías ágiles (Scrum, Kanban, Lean).
- Recomendaciones para la adopción y uso eficiente en diferentes escenarios.

Actividades

Actividad 1: Investigación y presentación sobre herramientas digitales ágiles

Objetivo: Identificar las principales herramientas digitales utilizadas en gestión ágil y diferenciar sus funciones y características.

Descripción:

- Dividir la clase en grupos pequeños y asignar a cada grupo una herramienta digital (por ejemplo, Jira, Trello, Asana, Slack, etc.).
- Cada grupo debe investigar las características, funcionalidades, ventajas y casos de uso de la herramienta asignada.
- Preparar una presentación breve (10 minutos) para compartir con el resto del grupo.
- Realizar una sesión plenaria donde cada grupo exponga y responda preguntas.

Organización: Grupos de 3-4 estudiantes

Producto esperado: Presentación multimedia y resumen escrito con características y funciones de la herramienta.

Duración estimada: 2 horas (investigación y presentación)

Actividad 2: Simulación práctica de gestión de un sprint usando Jira o Trello

Objetivo: Aplicar software de seguimiento y control para planificar y monitorear proyectos ágiles en escenarios simulados.

Descripción:

- El docente propone un proyecto de desarrollo de software sencillo, con un backlog inicial.
- Los estudiantes, en equipos, configuran el proyecto en Jira o Trello, creando tableros, listas, y tarjetas representando user stories y tareas.
- Planifican un sprint asignando tareas a los integrantes del equipo.
- Durante la simulación, actualizan el estado de las tareas diariamente para reflejar el avance.
- Al final, generan reportes de burndown y discuten los resultados.

Organización: Grupos de 4-5 estudiantes

Producto esperado: Tablero activo con backlog y sprint, reporte de avance y reflexión escrita sobre la experiencia.

Duración estimada: 3 horas (configuración, ejecución y reporte)

Actividad 3: Análisis de integración de herramientas para mejorar la colaboración

Objetivo: Analizar la integración de herramientas digitales en el proceso ágil para mejorar la colaboración y la eficiencia del equipo de desarrollo.

Descripción:

- Se presentan casos de estudio donde se integran herramientas de gestión con comunicación (por ejemplo, Jira + Slack).
- Los estudiantes analizan cómo la integración impacta la colaboración y la eficiencia.
- Discuten en grupos posibles mejoras y proponen flujos de trabajo integrados.
- Cada grupo presenta sus conclusiones y recomendaciones.

Organización: Grupos de 3-4 estudiantes

Producto esperado: Informe escrito y presentación de recomendaciones para integración.

Duración estimada: 2 horas

Actividad 4: Evaluación crítica y recomendación de herramientas para un proyecto dado

Objetivo: Evaluar críticamente el uso de diferentes herramientas ágiles según requisitos y contexto, proponiendo recomendaciones para su selección y uso adecuado.

Descripción:

- Se entrega a cada estudiante o grupo un perfil de proyecto con características específicas (equipo pequeño, remoto, con necesidades de integración, etc.).
- Los estudiantes deben seleccionar las herramientas digitales más adecuadas y justificar su elección considerando ventajas, limitaciones y contexto.
- Preparan un documento formal con la evaluación y recomendaciones de uso.
- Finalmente, se realiza una sesión de discusión grupal donde se comparan y analizan las diferentes propuestas.

Organización: Individual o grupos pequeños

Producto esperado: Informe crítico y recomendaciones de selección de herramientas.

Duración estimada: 2 horas

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre herramientas digitales en proyectos ágiles.

Cómo se evalúa: Cuestionario breve (quiz) con preguntas de opción múltiple y verdadero/falso sobre herramientas y conceptos ágiles.

Instrumento sugerido: Test en plataforma digital o papel, con 10 preguntas.

Evaluación formativa

Qué se evalúa: Progreso en la identificación, aplicación y análisis de herramientas digitales durante las actividades prácticas.

Cómo se evalúa: Observación directa, revisión de productos parciales (presentaciones, tableros configurados, informes de integración) y retroalimentación continua.

Instrumento sugerido: Rúbricas para presentación, uso de software y análisis crítico; listas de cotejo para participación en discusiones.

Evaluación sumativa

Qué se evalúa: Capacidad para evaluar críticamente y recomendar herramientas digitales adecuadas para un proyecto ágil específico.

Cómo se evalúa: Entrega y presentación de informe crítico con recomendaciones fundamentadas.

Instrumento sugerido: Rúbrica que valore claridad, profundidad del análisis, justificación basada en criterios y calidad de la comunicación escrita y oral.

Unidad 8: Planificación y Estimación en Proyectos de Software

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de analizar diferentes técnicas de planificación y estimación de proyectos de software, identificando sus ventajas y limitaciones en contextos reales.
- Al finalizar la unidad, el estudiante será capaz de aplicar métodos de estimación de tiempos y recursos, como Puntos de Función y Planning Poker, para elaborar planes de proyecto coherentes y realistas.
- Al finalizar la unidad, el estudiante será capaz de diseñar un plan de proyecto utilizando metodologías tradicionales y ágiles, adaptando las prácticas según los requerimientos y características del proyecto.
- Al finalizar la unidad, el estudiante será capaz de evaluar críticamente los resultados de la planificación y estimación, proponiendo ajustes para mejorar la precisión y efectividad en la gestión del proyecto.
- Al finalizar la unidad, el estudiante será capaz de comunicar de manera clara y estructurada los planes y estimaciones de proyectos de software, utilizando herramientas y técnicas adecuadas para contextos académicos y profesionales.

Contenidos Temáticos

1. Introducción a la planificación y estimación en proyectos de software

- Conceptos básicos de planificación y estimación: definición, importancia y objetivos.
- Impacto de una adecuada planificación en el éxito de proyectos de software.
- Factores que influyen la planificación y estimación: alcance, complejidad, recursos, equipo y tecnologías.

2. Técnicas y métodos de planificación y estimación

- Revisión de técnicas tradicionales y modernas de planificación.
 - Planificación en cascada (Waterfall).
 - Planificación iterativa e incremental.
 - Planificación en metodologías ágiles.
- Métodos de estimación de tiempos y recursos.
 - Puntos de Función: concepto, cálculo y aplicación práctica.
 - Planning Poker: dinámica, reglas y ventajas en estimación colaborativa.
 - Estimación por analogía y por juicio experto.
 - Estimación paramétrica y basada en métricas.
- Ventajas y limitaciones de cada técnica en contextos reales.

3. Diseño y elaboración de planes de proyecto

- Elementos esenciales de un plan de proyecto de software: alcance, cronograma, recursos, riesgos y calidad.
- Adaptación del plan a metodologías tradicionales y ágiles.
 - Planificación detallada frente a planificación adaptativa.
 - Herramientas para la gestión de planes (diagramas de Gantt, tableros Kanban, etc.).
- Integración de técnicas de estimación en el diseño del plan.

4. Evaluación y ajuste de la planificación y estimación

- Análisis crítico de resultados de planificación y estimación.
- Identificación de desviaciones y causas comunes en estimaciones.
- Métodos para ajustar y mejorar la precisión de la planificación.
 - Reestimación continua en proyectos ágiles.
 - Lecciones aprendidas y mejora continua.

5. Comunicación efectiva de planes y estimaciones

- Elementos clave para una comunicación clara y estructurada.
- Formatos y herramientas para presentar planes y estimaciones.
 - Documentación formal: informes, presentaciones y reportes.
 - Herramientas visuales: diagramas, gráficos y tableros colaborativos.
- Adaptación del mensaje según audiencia académica o profesional.

Actividades

Actividad 1: Análisis comparativo de técnicas de estimación

Objetivo: Contribuir al análisis crítico de diferentes técnicas de planificación y estimación, identificando ventajas y limitaciones.

Descripción paso a paso:

- Dividir a los estudiantes en grupos pequeños.
- Asignar a cada grupo una técnica específica (Puntos de Función, Planning Poker, estimación por analogía, etc.).
- Investigar la técnica asignada, sus fundamentos, proceso y casos de uso.
- Elaborar un cuadro comparativo que incluya ventajas, limitaciones y contextos de aplicación.
- Presentar los resultados al resto de la clase para discusión y retroalimentación.

Organización: Grupos de 3-4 estudiantes.

Producto esperado: Cuadro comparativo y presentación grupal.

Duración estimada: 2 horas.

Actividad 2: Estimación práctica usando Puntos de Función y Planning Poker

Objetivo: Aplicar métodos de estimación para elaborar planes de proyecto coherentes y realistas.

Descripción paso a paso:

- Proporcionar a los estudiantes un caso de estudio simple con requerimientos funcionales definidos.
- Individualmente, realizar la estimación usando Puntos de Función para calcular el tamaño del software.
- Formar grupos para aplicar Planning Poker y consensuar una estimación colectiva del esfuerzo en horas o días.
- Comparar ambas estimaciones y discutir las diferencias y beneficios de cada método.

Organización: Individual (estimación Puntos de Función) y grupos (Planning Poker).

Producto esperado: Documento con estimaciones individuales y grupales, con análisis comparativo.

Duración estimada: 3 horas.

Actividad 3: Diseño de un plan de proyecto adaptado a metodologías tradicionales y ágiles

Objetivo: Diseñar un plan de proyecto utilizando metodologías tradicionales y ágiles, adaptando prácticas según el proyecto.

Descripción paso a paso:

- Dividir la clase en dos grupos; uno diseñará un plan usando metodología tradicional y el otro usando metodología ágil.
- Proporcionar un mismo caso de proyecto para ambos grupos.
- Cada grupo elaborará un plan que incluya alcance, cronograma, asignación de recursos y gestión de riesgos.
- Presentar los planes y realizar una discusión comparativa sobre las diferencias, ventajas y desafíos de cada enfoque.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Plan de proyecto formal para cada metodología y presentación con análisis comparativo.

Duración estimada: 4 horas.

Actividad 4: Evaluación y propuesta de ajustes a una planificación existente

Objetivo: Evaluar críticamente los resultados de la planificación y estimación, proponiendo ajustes para mejorar precisión y efectividad.

Descripción paso a paso:

- Proporcionar a los estudiantes un caso con una planificación y estimación de proyecto ya elaborada, con desviaciones identificadas.
- En parejas, analizar las posibles causas de desviaciones y errores en la estimación.
- Proponer un plan de ajustes para mejorar la planificación en futuras iteraciones.
- Compartir las propuestas con el grupo para discusión y retroalimentación.

Organización: Parejas.

Producto esperado: Informe breve con análisis y propuestas de ajuste.

Duración estimada: 2 horas.

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre técnicas y conceptos básicos de planificación y estimación en proyectos de software.

Cómo se evalúa: Cuestionario breve con preguntas de opción múltiple y preguntas abiertas.

Instrumento sugerido: Test en línea o en papel con 10 preguntas sobre definiciones, técnicas y conceptos clave.

Evaluación formativa

Qué se evalúa: Aplicación práctica de técnicas de estimación, análisis crítico y diseño de planes.

Cómo se evalúa: Revisión continua y retroalimentación de las actividades prácticas (cuadros comparativos, estimaciones, planes, propuestas de ajuste).

Instrumento sugerido: Rúbrica de evaluación para cada actividad considerando precisión, coherencia, análisis crítico y presentación.

Evaluación sumativa

Qué se evalúa: Integración de conocimientos y habilidades para diseñar, evaluar y comunicar planes y estimaciones de proyectos de software.

Cómo se evalúa: Proyecto final individual o en parejas donde el estudiante debe elaborar un plan de proyecto completo con estimaciones, análisis crítico y presentación formal.

Instrumento sugerido: Rúbrica que considere calidad técnica del plan, precisión en estimaciones, capacidad crítica y claridad en la comunicación escrita y oral.

Unidad 9: Gestión de Riesgos y Calidad en el Desarrollo de Software

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar y analizar los principales riesgos en proyectos de desarrollo de software, aplicando técnicas sistemáticas para su evaluación y priorización.
- Al finalizar la unidad, el estudiante será capaz de diseñar planes de gestión de riesgos que integren estrategias preventivas y correctivas, considerando las características específicas del proyecto y su entorno.
- Al finalizar la unidad, el estudiante será capaz de aplicar métodos y herramientas para asegurar la calidad en el ciclo de vida del software, evaluando la eficacia de los procesos y entregables.
- Al finalizar la unidad, el estudiante será capaz de evaluar críticamente diferentes estándares y métricas de calidad en proyectos de software, proponiendo mejoras basadas en análisis de resultados.

- Al finalizar la unidad, el estudiante será capaz de comunicar efectivamente propuestas y resultados relacionados con la gestión de riesgos y aseguramiento de calidad en contextos académicos y profesionales.

Contenidos Temáticos

1. Introducción a la Gestión de Riesgos en Proyectos de Software

- Definición y importancia de la gestión de riesgos en el desarrollo de software
- Tipos de riesgos en proyectos de software: técnicos, organizacionales, externos y de proyecto
- Impacto de los riesgos no gestionados en el éxito del proyecto

2. Identificación y Análisis de Riesgos en Proyectos de Software

- Técnicas para la identificación de riesgos: brainstorming, listas de verificación, entrevistas, análisis histórico
- Herramientas para documentación de riesgos: registros de riesgos (Risk Register)
- Métodos para el análisis cualitativo de riesgos: evaluación de probabilidad e impacto
- Métodos para el análisis cuantitativo de riesgos: cálculo de valor monetario esperado, análisis de sensibilidad y simulaciones
- Priorización de riesgos basándose en el análisis

3. Diseño de Planes de Gestión de Riesgos

- Elementos de un plan de gestión de riesgos
- Estrategias preventivas: mitigación, transferencia, aceptación y evitación
- Estrategias correctivas y de respuesta ante riesgos materializados
- Integración del plan de riesgos con el plan general del proyecto
- Asignación de responsabilidades y recursos para la gestión de riesgos

4. Aseguramiento de la Calidad en el Ciclo de Vida del Software

- Conceptos fundamentales de calidad de software y aseguramiento de calidad (QA)
- Procesos y actividades clave en QA: revisión de código, pruebas, auditorías y control de calidad
- Métodos de prueba: pruebas unitarias, de integración, sistema, aceptación y pruebas automatizadas
- Herramientas para la gestión y automatización de pruebas y control de calidad
- Evaluación de la eficacia de procesos y entregables mediante métricas

5. Estándares y Métricas de Calidad en Proyectos de Software

- Principales estándares aplicables: ISO/IEC 9126, ISO/IEC 25010, CMMI, IEEE 829
- Tipos de métricas de calidad: métricas de producto, métricas de proceso y métricas de proyecto
- Interpretación y uso de métricas para la mejora continua
- Análisis crítico de resultados y propuestas de mejora basadas en métricas y estándares

6. Comunicación Efectiva de Gestión de Riesgos y Calidad

- Importancia de la comunicación en la gestión de riesgos y calidad
- Elaboración de reportes de riesgos y calidad para audiencias técnicas y no técnicas
- Técnicas para presentación de resultados y propuestas
- Casos prácticos y simulaciones para comunicar hallazgos y recomendaciones

Actividades

Actividad 1: Identificación y Análisis de Riesgos en un Proyecto Simulado

Objetivo: Identificar y analizar riesgos aplicando técnicas sistemáticas (Objetivo 1)

Descripción:

- Se presenta a los estudiantes un caso de estudio de un proyecto de desarrollo de software.
- En grupos, realizan una sesión de brainstorming para identificar posibles riesgos.
- Utilizan una matriz de probabilidad e impacto para analizar y priorizar los riesgos identificados.
- Documentan los riesgos priorizados en un registro de riesgos.

Organización: Grupos de 4-5 estudiantes

Producto esperado: Registro de riesgos con análisis cualitativo y priorización

Duración estimada: 2 horas

Actividad 2: Diseño de un Plan de Gestión de Riesgos para un Proyecto Real

Objetivo: Diseñar planes de gestión de riesgos con estrategias preventivas y correctivas (Objetivo 2)

Descripción:

- Seleccionan o se les asigna un proyecto real o hipotético de desarrollo de software.
- Con base en el registro de riesgos, diseñan un plan de gestión que incluya estrategias específicas para cada riesgo.
- Definen responsables, recursos y métodos de seguimiento para la gestión de riesgos.
- Presentan el plan ante el grupo para retroalimentación.

Organización: Parejas o grupos pequeños (2-3 estudiantes)

Producto esperado: Documento de plan de gestión de riesgos

Duración estimada: 3 horas

Actividad 3: Evaluación y Aplicación de Métodos de Aseguramiento de Calidad

Objetivo: Aplicar métodos y herramientas para asegurar la calidad y evaluar eficacia (Objetivo 3)

Descripción:

- Se entrega un módulo de software con documentación y casos de prueba.
- Los estudiantes ejecutan pruebas unitarias y de integración usando herramientas seleccionadas.

- Registran defectos encontrados y evalúan la cobertura de pruebas.
- Discuten la eficacia de los procesos aplicados y proponen mejoras.

Organización: Individual o parejas

Producto esperado: Informe de pruebas y evaluación de calidad

Duración estimada: 3 horas

Actividad 4: Análisis Crítico de Estándares y Métricas de Calidad

Objetivo: Evaluar estándares y métricas, proponiendo mejoras basadas en análisis (Objetivo 4)

Descripción:

- Se proporcionan datos de métricas de calidad y descripciones de estándares aplicados en un proyecto.
- Los estudiantes analizan los resultados y discuten su adecuación y limitaciones.
- Elaboran un reporte con propuestas de mejora fundamentadas en el análisis.

Organización: Grupos de 3-4 estudiantes

Producto esperado: Reporte crítico con propuestas de mejora

Duración estimada: 2 horas

Actividad 5: Presentación Oral de Resultados y Propuestas de Gestión de Riesgos y Calidad

Objetivo: Comunicar efectivamente propuestas y resultados (Objetivo 5)

Descripción:

- Cada grupo presenta ante la clase los resultados de su plan de gestión de riesgos y/o evaluaciones de calidad.
- Se enfatiza claridad, adecuación del lenguaje para la audiencia y uso de soporte visual.
- Se realiza una sesión de preguntas y retroalimentación.

Organización: Grupos o parejas según actividades previas

Producto esperado: Presentación oral con soporte visual

Duración estimada: 1.5 horas

Evaluación

Evaluación Diagnóstica

Qué se evalúa: Conocimientos previos sobre riesgos y calidad en proyectos de software.

Cómo se evalúa: Cuestionario breve con preguntas abiertas y de opción múltiple.

Instrumento sugerido: Test en línea o en papel con 10-15 preguntas.

Evaluación Formativa

Qué se evalúa: Progreso en la identificación, análisis, diseño e implementación de gestión de riesgos y aseguramiento de calidad.

Cómo se evalúa: Revisión continua de productos parciales (registro de riesgos, planes, informes de pruebas) y retroalimentación en actividades grupales e individuales.

Instrumento sugerido: Listas de cotejo para cada actividad, rúbricas de evaluación para informes y presentaciones.

Evaluación Sumativa

Qué se evalúa: Competencia integral para identificar riesgos, diseñar planes, aplicar métodos de calidad, analizar métricas y comunicar resultados.

Cómo se evalúa: Proyecto final que incluya un análisis completo de riesgos, plan de gestión, aplicación de técnicas de aseguramiento de calidad, análisis crítico de métricas y presentación oral.

Instrumento sugerido: Rúbrica de evaluación que considere: calidad técnica, profundidad del análisis, aplicabilidad, claridad en la comunicación y colaboración.

Unidad 10: Integración Continua y Entrega Continua

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de explicar los conceptos fundamentales de integración continua (CI) y entrega continua (CD) identificando sus beneficios en la mejora de la eficiencia y calidad del desarrollo de software.
- Al finalizar la unidad, el estudiante será capaz de configurar y ejecutar pipelines básicos de CI/CD utilizando herramientas populares, aplicando buenas prácticas para automatizar pruebas y despliegues.
- Al finalizar la unidad, el estudiante será capaz de analizar y evaluar diferentes estrategias de CI/CD para seleccionar la más adecuada según el contexto específico de un proyecto de software.
- Al finalizar la unidad, el estudiante será capaz de integrar técnicas de CI/CD en metodologías ágiles y tradicionales, demostrando cómo estas prácticas contribuyen a la mejora continua del ciclo de vida del desarrollo de software.

Contenidos Temáticos

1. Introducción a la Integración Continua (CI) y Entrega Continua (CD)

- **Conceptos fundamentales de CI y CD:** Definición, diferencias y relación entre integración continua, entrega continua y despliegue continuo.
- **Historia y evolución:** Origen de CI/CD y su evolución en el contexto del desarrollo de software moderno.
- **Beneficios de CI/CD:** Cómo CI/CD contribuye a la mejora de la eficiencia, calidad, detección temprana de errores y reducción del tiempo de entrega.
- **Terminología clave:** Pipelines, builds, tests automatizados, artefactos, rollback, entre otros términos relevantes.

2. Herramientas y Tecnologías para CI/CD

- **Herramientas populares para CI/CD:** Jenkins, GitLab CI, Travis CI, CircleCI, GitHub Actions, Azure DevOps.

- **Repositorios de código y su integración con CI/CD:** Uso de Git, GitHub, GitLab, Bitbucket en pipelines.
- **Automatización de pruebas:** Tipos de pruebas automatizadas (unitarias, integración, funcionales) y su integración en pipelines.
- **Contenedores y entornos:** Uso de Docker y Kubernetes para despliegues automatizados.

3. Configuración y Ejecución de Pipelines Básicos de CI/CD

- **Diseño de pipelines:** Etapas comunes: build, test, deploy.
- **Creación de un pipeline básico:** Configuración paso a paso en una herramienta seleccionada (ejemplo: GitHub Actions o Jenkins).
- **Integración de pruebas automatizadas:** Cómo incluir pruebas unitarias y de integración en la pipeline.
- **Despliegue automatizado:** Estrategias básicas de despliegue: despliegue en entornos de prueba, staging y producción.
- **Buenas prácticas en pipelines:** Manejo de secretos, notificaciones, rollback y monitoreo.

4. Estrategias de CI/CD y Evaluación Contextual

- **Estrategias de integración continua:** Integración frecuente, feature branching, trunk-based development.
- **Estrategias de entrega continua y despliegue continuo:** Entrega manual vs automática, despliegues canary, blue-green, rolling updates.
- **Factores para seleccionar estrategias:** Tamaño del equipo, tipo de proyecto, frecuencia de cambios, requisitos de calidad y seguridad.
- **Evaluación comparativa:** Análisis de casos reales y estudios de caso para seleccionar la estrategia adecuada.

5. Integración de CI/CD en Metodologías Ágiles y Tradicionales

- **CI/CD en metodologías ágiles:** Scrum, Kanban y su relación con la automatización de pipelines.
- **CI/CD en metodologías tradicionales:** Adaptación de prácticas CI/CD en modelos en cascada y otros.
- **Mejora continua y feedback:** Cómo CI/CD facilita ciclos de retroalimentación rápidos y mejora continua.
- **Rol del equipo de desarrollo y operaciones:** Cultura DevOps, colaboración y responsabilidades en equipos integrados.

Actividades

Actividad 1: Debate y Análisis de Beneficios de CI/CD

Objetivo: Explicar los conceptos fundamentales de CI/CD e identificar sus beneficios en el desarrollo de software.

Descripción:

- Dividir la clase en grupos de 4-5 estudiantes.
- Cada grupo investigará un beneficio específico de CI/CD (por ejemplo, reducción de errores, rapidez en entregas, mejora de la calidad).

- Preparar una breve presentación para explicar el beneficio y ejemplos prácticos.
- Realizar un debate guiado donde se discutan las presentaciones y se contrasten puntos de vista.

Organización: Grupos

Producto esperado: Presentación y resumen escrito de los beneficios discutidos.

Duración estimada: 90 minutos

Actividad 2: Configuración Práctica de un Pipeline Básico de CI/CD

Objetivo: Configurar y ejecutar pipelines básicos usando herramientas populares aplicando buenas prácticas.

Descripción:

- El docente proporcionará un repositorio base con código fuente simple (ejemplo: aplicación web o script).
- Los estudiantes configurarán un pipeline en GitHub Actions o Jenkins que incluya build, pruebas unitarias y despliegue a un entorno de prueba (local o en la nube).
- Se guiará paso a paso para incluir notificaciones y manejo básico de errores.
- Al final, cada estudiante ejecutará el pipeline y documentará el proceso y resultados.

Organización: Individual o parejas

Producto esperado: Pipeline configurado y documentación del proceso.

Duración estimada: 3 horas

Actividad 3: Análisis Comparativo de Estrategias CI/CD para Casos Prácticos

Objetivo: Analizar y evaluar diferentes estrategias de CI/CD para seleccionar la más adecuada según el contexto de un proyecto.

Descripción:

- Se entregarán a los grupos varios casos de proyectos con características distintas (equipo pequeño, proyecto crítico, equipo distribuido, etc.).
- Cada grupo evaluará las estrategias de CI/CD más adecuadas para su caso, justificando la elección.
- Prepararán un informe y presentación con su análisis y recomendaciones.
- Se realizará una sesión de preguntas y retroalimentación en clase.

Organización: Grupos

Producto esperado: Informe y presentación de estrategia seleccionada.

Duración estimada: 2 horas

Actividad 4: Integración de CI/CD en un Proyecto Ágil - Simulación

Objetivo: Integrar técnicas de CI/CD en metodologías ágiles, demostrando su impacto en la mejora continua del ciclo de vida.

Descripción:

- Formar equipos que simulen un proyecto ágil con roles definidos (Scrum Master, desarrolladores, testers).
- Planificar un sprint corto donde se debe implementar una funcionalidad con pipeline CI/CD configurado para pruebas y despliegue.
- Durante la simulación, los estudiantes deberán documentar cómo CI/CD facilita la entrega y retroalimentación.
- Finalizar con una reflexión grupal sobre las ventajas y desafíos encontrados.

Organización: Grupos

Producto esperado: Reporte de simulación y presentación de conclusiones.

Duración estimada: 3 horas

Evaluación

Evaluación Diagnóstica

Qué se evalúa: Conocimientos previos sobre prácticas y conceptos básicos de CI/CD.

Cómo se evalúa: Cuestionario de opción múltiple y preguntas abiertas breves.

Instrumento sugerido: Test en plataforma virtual o en papel al inicio de la unidad.

Evaluación Formativa

Qué se evalúa: Progreso en la comprensión y aplicación práctica de conceptos mediante actividades y retroalimentación continua.

Cómo se evalúa: Revisión de avances en actividades prácticas, participación en debates, calidad de documentación y presentación de trabajos.

Instrumento sugerido: Rúbricas para actividades prácticas, observación directa y autoevaluación entre pares.

Evaluación Sumativa

Qué se evalúa: Dominio integral de conceptos, habilidades para configurar pipelines, análisis crítico de estrategias y aplicación en metodologías.

Cómo se evalúa: Proyecto final que incluya:

- Configuración completa de un pipeline básico de CI/CD.
- Análisis escrito justificando la estrategia seleccionada para un caso real o simulado.
- Reflexión sobre la integración de CI/CD en metodologías ágiles o tradicionales.

Instrumento sugerido: Rúbrica detallada de proyecto final, presentación oral y entrega de documentación.

Unidad 11: Métricas y Evaluación de Proyectos de Software

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar y describir las principales métricas utilizadas para el seguimiento y evaluación de proyectos de software, aplicando criterios de selección adecuados según el contexto

del proyecto.

- Al finalizar la unidad, el estudiante será capaz de analizar datos derivados de métricas de proyectos de software para evaluar el desempeño y calidad del proyecto, fundamentando sus conclusiones en evidencia cuantitativa.
- Al finalizar la unidad, el estudiante será capaz de aplicar técnicas de medición y evaluación para monitorear el progreso de un proyecto de software, utilizando herramientas específicas para facilitar la toma de decisiones basadas en datos.
- Al finalizar la unidad, el estudiante será capaz de diseñar un plan de métricas y evaluación para un proyecto de software, integrando indicadores que permitan proponer mejoras continuas en el proceso de desarrollo.

Contenidos Temáticos

1. Introducción a las métricas en proyectos de software

- Definición y propósito de las métricas en el desarrollo de software: importancia para el control, mejora y toma de decisiones.
- Clasificación de métricas: métricas de producto, proceso y proyecto.
- Contexto y criterios para la selección adecuada de métricas según el tipo de proyecto y metodología de desarrollo.

2. Principales métricas para el seguimiento y evaluación de proyectos de software

- Métricas de tamaño y complejidad: líneas de código (LOC), puntos de función, complejidad ciclomática.
- Métricas de calidad del producto: densidad de defectos, tasa de fallos, cobertura de pruebas, calidad del código.
- Métricas de desempeño del proyecto: avance frente a plan, tasa de cumplimiento de hitos, esfuerzo consumido, velocidad (en metodologías ágiles).
- Métricas de productividad: esfuerzo por unidad de tamaño, productividad del equipo.
- Métricas de satisfacción y calidad percibida: feedback del cliente, índice de satisfacción, tasa de uso.

3. Análisis e interpretación de datos derivados de métricas

- Técnicas para la recopilación y organización de datos métricos.
- Uso de análisis estadístico básico para interpretar métricas: promedios, tendencias, desviaciones estándar.
- Identificación de patrones y desviaciones en el desempeño y calidad del proyecto.
- Elaboración de informes basados en evidencia cuantitativa para la toma de decisiones.

4. Técnicas y herramientas para la medición y evaluación del progreso

- Instrumentos y herramientas comunes para la medición automatizada: sistemas de control de versiones, herramientas de gestión de proyectos, software de análisis estático.
- Integración de métricas en tableros de control (dashboards) para el monitoreo continuo.
- Prácticas para la evaluación periódica y el ajuste del plan de métricas.

5. Diseño de un plan de métricas y evaluación para proyectos de software

- Definición de objetivos de medición alineados con metas del proyecto.
- Selección y justificación de métricas pertinentes según contexto y ciclo de vida del proyecto.
- Planificación de la recopilación de datos y definición de responsables.
- Propuesta de indicadores clave de desempeño (KPIs) y mecanismos para su seguimiento.
- Uso del plan para identificar áreas de mejora continua y retroalimentación al equipo.

Actividades

Actividad 1: Identificación y clasificación de métricas en un caso de estudio

Objetivo: Contribuye al primer objetivo: identificar y describir las principales métricas y aplicar criterios de selección.

Descripción:

- Se presenta a los estudiantes un caso real o simulado de un proyecto de software.
- En grupos, analizan el proyecto para identificar qué métricas serían relevantes para su seguimiento y evaluación.
- Cada grupo clasifica las métricas en producto, proceso y proyecto, y justifica la selección según el contexto.
- Se realiza una discusión plenaria para comparar los criterios y conclusiones de cada grupo.

Organización: Grupos de 3-4 estudiantes.

Producto esperado: Documento con lista clasificada de métricas y justificación contextual.

Duración estimada: 90 minutos.

Actividad 2: Análisis cuantitativo de métricas y elaboración de informe

Objetivo: Contribuye al segundo objetivo: analizar datos métricos para evaluar desempeño y calidad.

Descripción:

- Se entrega un conjunto de datos métricos (por ejemplo, defectos reportados, esfuerzo, velocidad) de un proyecto.
- Individualmente, cada estudiante realiza cálculos estadísticos básicos y gráficos para interpretar las métricas.
- Elaboran un informe que describa el estado del proyecto y posibles riesgos o áreas a mejorar, con base en la evidencia cuantitativa.

Organización: Individual.

Producto esperado: Informe de análisis con conclusiones fundamentadas.

Duración estimada: 120 minutos.

Actividad 3: Uso de herramientas para monitoreo de métricas en proyectos

Objetivo: Contribuye al tercer objetivo: aplicar técnicas y herramientas para monitorear el progreso.

Descripción:

- Se introduce una herramienta de gestión de proyectos o análisis de código (por ejemplo, Jira, SonarQube, o similar).
- En parejas, los estudiantes configuran un tablero de métricas con indicadores relevantes para un proyecto dado.
- Simulan la actualización periódica de métricas y discuten cómo estas facilitan la toma de decisiones.

Organización: Parejas.

Producto esperado: Capturas o exportación del tablero configurado y breve presentación oral.

Duración estimada: 90 minutos.

Actividad 4: Diseño de un plan de métricas y evaluación para un proyecto

Objetivo: Contribuye al cuarto objetivo: diseñar un plan de métricas que permita mejoras continuas.

Descripción:

- En grupos, se les asigna un tipo de proyecto o contexto específico (por ejemplo, desarrollo ágil, proyecto con alta criticidad).
- Diseñan un plan de métricas que incluya objetivos, métricas seleccionadas, responsables, periodicidad y mecanismos de reporte.
- Presentan su plan a la clase y reciben retroalimentación para mejorar su propuesta.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Documento del plan de métricas y presentación oral.

Duración estimada: 2 horas.

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre métricas de software y familiaridad con conceptos básicos de evaluación de proyectos.

Cómo se evalúa: Cuestionario breve con preguntas de opción múltiple y verdadero/falso sobre definiciones y tipos de métricas.

Instrumento sugerido: Test en línea o papel con 10-15 preguntas.

Evaluación formativa

Qué se evalúa: Progreso en la identificación, análisis y aplicación de métricas; comprensión de las técnicas y herramientas.

Cómo se evalúa: Revisión de productos parciales de actividades (listas de métricas, informes de análisis, tableros configurados).

Instrumento sugerido: Rúbricas específicas para cada actividad que valoren precisión, justificación, análisis y aplicación práctica.

Evaluación sumativa

Qué se evalúa: Capacidad para diseñar un plan integral de métricas y evaluación, fundamentar decisiones con datos y comunicar resultados.

Cómo se evalúa: Entrega final del plan de métricas con justificación, presentación oral del plan y defensa ante preguntas.

Instrumento sugerido: Rúbrica que incluya criterios de selección adecuada de métricas, coherencia del plan, uso de evidencia, calidad de presentación y argumentación.

Unidad 12: Adaptación y Selección de Metodologías según el Contexto

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de analizar las características y requisitos de diferentes proyectos de software para seleccionar la metodología de desarrollo más adecuada según el contexto.
- Al finalizar la unidad, el estudiante será capaz de evaluar críticamente las ventajas y limitaciones de diversas metodologías de desarrollo en escenarios específicos, justificando la selección propuesta.
- Al finalizar la unidad, el estudiante será capaz de adaptar prácticas y procesos de metodologías tradicionales y ágiles para satisfacer las necesidades particulares de proyectos con diferentes restricciones y objetivos.
- Al finalizar la unidad, el estudiante será capaz de diseñar un plan de implementación de la metodología seleccionada, integrando criterios de calidad y gestión de proyectos para optimizar el ciclo de vida del desarrollo.
- Al finalizar la unidad, el estudiante será capaz de comunicar de forma clara y fundamentada la elección y adaptación de metodologías a audiencias académicas y profesionales.

Contenidos Temáticos

1. Introducción a la Adaptación y Selección de Metodologías

- Contexto y relevancia de elegir metodologías adecuadas para proyectos de software.
- Factores clave del entorno que influyen en la selección metodológica.
- Resumen de metodologías tradicionales vs. ágiles y sus características básicas.

2. Análisis de Características y Requisitos de Proyectos de Software

- Identificación y análisis de requisitos funcionales y no funcionales.
- Evaluación del tamaño, complejidad y alcance del proyecto.
- Consideraciones sobre el equipo de trabajo, experiencia y cultura organizacional.
- Restricciones de tiempo, presupuesto y recursos.

3. Criterios para la Selección de Metodologías de Desarrollo

- Evaluación de la flexibilidad y adaptabilidad requerida.
- Impacto de la incertidumbre y cambios en los requisitos.
- Necesidades de documentación y control de calidad.
- Compatibilidad con herramientas y tecnologías disponibles.

4. Evaluación Crítica de Ventajas y Limitaciones de Metodologías

- Análisis comparativo entre metodologías tradicionales (Waterfall, V-Model).
- Análisis comparativo entre metodologías ágiles (Scrum, Kanban, XP).
- Escenarios típicos de aplicación y casos de éxito/fallo.
- Discusión sobre riesgos y mitigación en la selección metodológica.

5. Adaptación de Prácticas y Procesos Metodológicos

- Modificación y combinación de prácticas tradicionales y ágiles.
- Ajuste de procesos para proyectos con restricciones específicas.
- Integración de prácticas de aseguramiento de calidad y gestión de riesgos.
- Ejemplos prácticos de adaptaciones metodológicas en distintos tipos de proyectos.

6. Diseño de un Plan de Implementación de la Metodología Seleccionada

- Estructura y componentes esenciales del plan de implementación.
- Definición de roles, responsabilidades y flujos de trabajo.
- Incorporación de criterios de calidad y métricas de desempeño.
- Planificación de capacitación y gestión del cambio organizacional.

7. Comunicación y Justificación de la Selección y Adaptación Metodológica

- Técnicas para la presentación clara y fundamentada de decisiones metodológicas.
- Uso de evidencia y análisis crítico para justificar la selección y adaptación.
- Preparación de informes, presentaciones y documentos técnicos.
- Buenas prácticas para la comunicación con audiencias técnicas y no técnicas.

Actividades

1. Análisis de Caso: Selección Metodológica según Requisitos del Proyecto

Objetivo: Analizar características y requisitos para seleccionar la metodología adecuada.

Descripción:

- Se presenta un caso realista con detalles de un proyecto de software (requisitos, equipo, restricciones).
- Los estudiantes revisan la información y discuten en grupo las necesidades del proyecto.
- Cada grupo propone una metodología de desarrollo justificando su elección con base en los criterios aprendidos.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Informe escrito con análisis y justificación de la selección metodológica.

Duración estimada: 2 horas.

2. Debate Crítico: Ventajas y Limitaciones de Metodologías en Escenarios Reales

Objetivo: Evaluar críticamente ventajas y limitaciones de diversas metodologías.

Descripción:

- Se asignan diferentes metodologías a grupos (Scrum, Waterfall, Kanban, XP, etc.).
- Cada grupo prepara argumentos sobre ventajas, limitaciones y escenarios ideales para su metodología.
- Se realiza un debate estructurado donde cada grupo expone y responde preguntas.

Organización: Grupos de 3-4 estudiantes.

Producto esperado: Presentación oral y documento resumen con conclusiones.

Duración estimada: 2 horas.

3. Taller de Adaptación Metodológica: Ajustando Procesos a Necesidades Específicas

Objetivo: Adaptar prácticas y procesos para satisfacer necesidades particulares de proyectos.

Descripción:

- Se presenta un proyecto con restricciones específicas (ej. equipo pequeño, alta incertidumbre, presupuesto limitado).
- Los estudiantes diseñan una adaptación de una metodología tradicional o ágil, combinando prácticas según el contexto.
- Se desarrolla un plan de actividades y procesos adaptados que incluyen aseguramiento de calidad.

Organización: Grupos de 4 estudiantes.

Producto esperado: Documento con propuesta de adaptación metodológica y justificación.

Duración estimada: 3 horas.

4. Diseño y Presentación de un Plan de Implementación Metodológica

Objetivo: Diseñar un plan que integre criterios de calidad y gestión para la implementación metodológica.

Descripción:

- Cada estudiante o grupo elige una metodología para un proyecto ficticio o real.
- Elaboran un plan detallado que incluya roles, flujos de trabajo, métricas y estrategias de gestión del cambio.
- Preparan una presentación formal para comunicar su plan a una audiencia académica o profesional.

Organización: Individual o grupos de 2 estudiantes.

Producto esperado: Documento del plan de implementación y presentación oral.

Duración estimada: 4 horas.

Evaluación

Evaluación Diagnóstica

Qué se evalúa: Conocimientos previos sobre metodologías de desarrollo y criterios básicos de selección.

Cómo se evalúa: Cuestionario corto de opción múltiple y preguntas abiertas sobre conceptos clave.

Instrumento sugerido: Test en línea o en papel con 10 preguntas.

Evaluación Formativa

Qué se evalúa: Progreso en análisis, argumentación crítica, adaptación y diseño de planes metodológicos durante las actividades.

Cómo se evalúa: Revisión y retroalimentación continua de informes, debates, talleres y presentaciones.

Instrumento sugerido: Rúbricas detalladas para cada actividad que consideren análisis crítico, justificación, creatividad y comunicación.

Evaluación Sumativa

Qué se evalúa: Capacidad integral para seleccionar, adaptar, planificar y comunicar metodologías de desarrollo según contexto.

Cómo se evalúa: Examen escrito y entrega final de un proyecto que incluya análisis, adaptación y plan de implementación con justificación.

Instrumento sugerido: Examen estructurado y rúbrica para evaluación del proyecto final (documento y presentación).

Unidad 13: Trabajo en Equipo y Comunicación en Proyectos Ágiles

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de analizar las dinámicas de trabajo en equipo dentro de proyectos ágiles identificando roles y responsabilidades para mejorar la colaboración efectiva.
- Al finalizar la unidad, el estudiante será capaz de aplicar técnicas de comunicación verbal y escrita para facilitar la interacción y coordinación en equipos que utilizan metodologías ágiles.
- Al finalizar la unidad, el estudiante será capaz de diseñar estrategias de gestión de conflictos y toma de decisiones en equipos ágiles para optimizar el rendimiento grupal.
- Al finalizar la unidad, el estudiante será capaz de evaluar herramientas y prácticas colaborativas digitales que soportan la comunicación y el trabajo en equipo en entornos ágiles, seleccionando las más adecuadas según el contexto del proyecto.
- Al finalizar la unidad, el estudiante será capaz de integrar habilidades de trabajo en equipo y comunicación para presentar resultados y avances de proyectos ágiles de manera clara y efectiva ante distintos públicos.

Contenidos Temáticos

1. Dinámicas de Trabajo en Equipo en Proyectos Ágiles

- **1.1. Fundamentos del trabajo en equipo ágil:** Introducción a las características y beneficios del trabajo colaborativo en entornos ágiles. Importancia de la autoorganización y colaboración continua.

- **1.2. Roles y responsabilidades en equipos ágiles:** Definición y análisis de los roles clave (Scrum Master, Product Owner, Developers, etc.). Responsabilidades y expectativas para cada rol y cómo contribuyen a la colaboración efectiva.
- **1.3. Dinámicas y prácticas para fortalecer el equipo:** Técnicas ágiles como Daily Stand-ups, Retrospectivas, Pair Programming, y cómo fomentan la cohesión y productividad del equipo.

2. Técnicas de Comunicación en Equipos Ágiles

- **2.1. Comunicación verbal efectiva:** Estrategias para mejorar la claridad, escucha activa, feedback constructivo y reuniones efectivas en el contexto ágil.
- **2.2. Comunicación escrita ágil:** Uso de herramientas y formatos para documentación ligera, comunicación clara y concisa (ej. user stories, tickets, correos, chats).
- **2.3. Facilitación de la interacción y coordinación:** Técnicas para garantizar que la comunicación fluya entre todos los miembros del equipo y con stakeholders, uso de tableros Kanban, pizarras digitales.

3. Gestión de Conflictos y Toma de Decisiones en Equipos Ágiles

- **3.1. Identificación y análisis de conflictos comunes:** Tipos de conflictos en proyectos ágiles y sus causas frecuentes.
- **3.2. Estrategias para la resolución de conflictos:** Métodos colaborativos para resolver desacuerdos, negociación, mediación y consenso.
- **3.3. Modelos y técnicas de toma de decisiones:** Técnicas ágiles para la toma de decisiones rápida y efectiva (p.ej. votación, consenso, Delegated Decision Making).

4. Herramientas y Prácticas Colaborativas Digitales en Entornos Ágiles

- **4.1. Evaluación de herramientas digitales:** Análisis comparativo de herramientas como Jira, Trello, Slack, Microsoft Teams, Miro y su aplicabilidad según tipo de proyecto y equipo.
- **4.2. Prácticas colaborativas digitales efectivas:** Uso de tableros compartidos, videoconferencias, chats, documentación compartida y control de versiones para potenciar la colaboración.
- **4.3. Selección adecuada de herramientas y prácticas:** Criterios para elegir y adaptar las herramientas en función del contexto, tamaño del equipo, modalidad de trabajo (presencial/remoto/híbrido).

5. Integración de Habilidades para Presentación de Resultados en Proyectos Ágiles

- **5.1. Técnicas para estructurar presentaciones claras y efectivas:** Planificación de contenido, mensajes clave, uso de soportes visuales.
- **5.2. Comunicación orientada a distintos públicos:** Adaptación del lenguaje y enfoque para stakeholders técnicos, clientes y equipo.
- **5.3. Prácticas para la presentación de avances y resultados:** Presentación de Sprint Reviews, demos, informes breves y retroalimentación efectiva.

Actividades

Actividad 1: Mapeo de Roles y Responsabilidades en un Equipo Ágil

Objetivo: Analizar las dinámicas de trabajo en equipo identificando roles y responsabilidades para mejorar la colaboración efectiva.

Descripción:

- Se presenta un caso práctico con un proyecto de desarrollo ágil ficticio.
- Los estudiantes en grupos deben identificar y asignar roles ágiles a cada miembro del equipo, describiendo sus responsabilidades.
- Luego, discuten cómo esas responsabilidades afectan la colaboración y proponen mejoras para optimizar el trabajo en equipo.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Documento o presentación con el mapeo de roles y propuesta de mejora.

Duración estimada: 1.5 horas.

Actividad 2: Simulación de Comunicación Ágil en Reuniones Diarias

Objetivo: Aplicar técnicas de comunicación verbal y escrita para facilitar la interacción y coordinación en equipos ágiles.

Descripción:

- Los estudiantes simulan una Daily Stand-up en grupos, donde cada participante comunica sus avances, impedimentos y planes.
- Se practica la escucha activa y feedback constructivo.
- Después, cada grupo redacta un breve resumen escrito de la reunión que refleje los puntos clave.

Organización: Grupos de 5 estudiantes.

Producto esperado: Grabación o registro de la simulación y resumen escrito de la reunión.

Duración estimada: 1 hora.

Actividad 3: Taller de Gestión de Conflictos y Toma de Decisiones

Objetivo: Diseñar estrategias de gestión de conflictos y toma de decisiones para optimizar el rendimiento grupal.

Descripción:

- Se plantea un escenario con un conflicto típico en un equipo ágil.
- Los grupos deben identificar el conflicto, analizar causas y proponer estrategias para su resolución.
- Posteriormente, deben aplicar una técnica de toma de decisiones para definir una solución consensuada.

Organización: Grupos de 4 estudiantes.

Producto esperado: Informe que describa el conflicto, estrategias propuestas y resultado de la toma de decisiones.

Duración estimada: 2 horas.

Actividad 4: Evaluación y Selección de Herramientas Colaborativas para un Proyecto Ágil

Objetivo: Evaluar herramientas colaborativas digitales y seleccionar las adecuadas según el contexto del proyecto.

Descripción:

- Se presenta un perfil de proyecto ágil con características específicas (tamaño del equipo, modalidad de trabajo, tipo de producto).
- En grupos, los estudiantes investigan y evalúan diferentes herramientas colaborativas.
- Elaboran una matriz comparativa y justifican su selección de herramientas para ese proyecto.

Organización: Grupos de 3-4 estudiantes.

Producto esperado: Matriz comparativa y justificación escrita.

Duración estimada: 1.5 horas.

Actividad 5: Presentación de Resultados de un Sprint

Objetivo: Integrar habilidades de trabajo en equipo y comunicación para presentar resultados y avances de proyectos ágiles de manera clara y efectiva.

Descripción:

- Los grupos preparan una presentación simulando un Sprint Review ante un público hipotético (docente y compañeros).
- Se debe incluir explicación de avances, demostración del producto y respuesta a preguntas.
- Se enfatiza la adaptación del mensaje a la audiencia y el uso adecuado de soportes visuales.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Presentación oral con apoyo visual y reporte breve.

Duración estimada: 2 horas.

Evaluación

Evaluación Diagnóstica

Qué se evalúa: Conocimientos previos sobre roles en equipos ágiles y habilidades comunicativas básicas.

Cómo se evalúa: Cuestionario breve con preguntas abiertas y de opción múltiple sobre conceptos fundamentales.

Instrumento sugerido: Test en línea o impreso al inicio de la unidad.

Evaluación Formativa

Qué se evalúa: Procesos de identificación de roles, aplicación de técnicas de comunicación, resolución de conflictos y uso de herramientas colaborativas.

- Revisión y retroalimentación de productos parciales de las actividades prácticas (mapas de roles, resúmenes de reuniones, informes de conflictos, matrices de evaluación).
- Observación de la participación en simulaciones y talleres.

Instrumento sugerido: Rúbrica para evaluación de trabajos en grupo y observación directa.

Evaluación Sumativa

Qué se evalúa: Capacidad integrada para analizar dinámicas, comunicar efectivamente, gestionar conflictos, seleccionar herramientas y presentar resultados en proyectos ágiles.

- Presentación final del Sprint Review (oral y escrita) evaluada con rúbrica que considere claridad, contenido, uso de técnicas de comunicación y adaptación al público.
- Informe final que integre análisis de roles, gestión de conflictos y selección de herramientas.

Instrumento sugerido: Rúbrica detallada para presentación y reporte escrito.

Unidad 14: Estudio de Casos Reales

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de analizar proyectos de software reales identificando las metodologías de desarrollo aplicadas y sus características principales.
- Al finalizar la unidad, el estudiante será capaz de comparar y contrastar las ventajas y desventajas de las metodologías utilizadas en los casos estudiados, fundamentando sus argumentos con evidencias del proyecto.
- Al finalizar la unidad, el estudiante será capaz de evaluar críticamente los resultados y procesos de los proyectos reales para identificar lecciones aprendidas y oportunidades de mejora.
- Al finalizar la unidad, el estudiante será capaz de sintetizar recomendaciones basadas en los casos analizados para la aplicación efectiva de metodologías en escenarios futuros.
- Al finalizar la unidad, el estudiante será capaz de comunicar de manera clara y estructurada el análisis y las conclusiones derivadas del estudio de casos reales en presentaciones escritas y orales.

Contenidos Temáticos

1. Introducción al análisis de casos reales en desarrollo de software

- Importancia del estudio de casos para entender metodologías
- Objetivos y enfoques para el análisis de proyectos reales
- Contextualización del entorno y factores del proyecto

2. Identificación y descripción de metodologías aplicadas en proyectos reales

- Revisión de metodologías comunes: cascada, ágil, scrum, kanban, XP, etc.
- Herramientas para identificar la metodología en un caso de estudio
- Características principales de cada metodología en la práctica

3. Análisis comparativo de ventajas y desventajas de las metodologías aplicadas

- Metodologías y su adecuación al contexto del proyecto
- Comparación basada en criterios: flexibilidad, calidad, tiempo, costo, comunicación
- Uso de evidencia documental y resultados del proyecto para fundamentar argumentos

4. Evaluación crítica de resultados y procesos de los proyectos

- Identificación de éxitos y fallas en la aplicación de metodologías
- Análisis de factores que influyeron en los resultados (equipo, comunicación, gestión)
- Interpretación de métricas y resultados del proyecto

5. Lecciones aprendidas y oportunidades de mejora

- Extracción de aprendizajes útiles para futuros proyectos
- Reconocimiento de patrones de éxito y error en el desarrollo
- Propuestas de mejora basadas en la evidencia del estudio de casos

6. Síntesis y comunicación de conclusiones

- Estructuración de informes escritos claros y coherentes
- Diseño y elaboración de presentaciones orales efectivas
- Técnicas para argumentar y defender análisis y recomendaciones

Actividades

Actividad 1: Análisis individual de un caso real de desarrollo de software

Objetivo: Analizar proyectos reales identificando metodologías y sus características principales.

Descripción:

- Se entrega a cada estudiante un caso documentado de un proyecto de software real (puede ser de empresas reconocidas o casos académicos).
- El estudiante lee y extrae información relevante para identificar la metodología aplicada.
- Elabora un resumen donde describa la metodología y sus características principales en el contexto del proyecto.

Organización: Individual

Producto esperado: Reporte escrito individual con identificación y descripción de la metodología.

Duración estimada: 2 horas

Actividad 2: Debate en grupos sobre ventajas y desventajas de las metodologías utilizadas

Objetivo: Comparar y contrastar ventajas y desventajas fundamentadas en evidencia del proyecto.

Descripción:

- Los estudiantes forman grupos de 4-5 integrantes.
- Cada grupo recibe dos casos con metodologías diferentes.

- Discuten y elaboran tablas comparativas que enumeren ventajas y desventajas de cada metodología según el caso.
- Preparan una argumentación basada en evidencias para exponer ante el resto de la clase.

Organización: Grupos pequeños

Producto esperado: Tabla comparativa y presentación oral grupal.

Duración estimada: 3 horas (incluye preparación y presentación)

Actividad 3: Evaluación crítica y propuesta de mejora para un proyecto real

Objetivo: Evaluar críticamente resultados y procesos para identificar lecciones aprendidas y oportunidades de mejora.

Descripción:

- En parejas, los estudiantes seleccionan un caso de estudio previamente analizado.
- Analizan la información para identificar éxitos, fallas y factores determinantes en el desarrollo.
- Formulan propuestas concretas de mejora para futuras aplicaciones de la metodología.
- Elaboran un informe que contenga evaluación crítica y recomendaciones.

Organización: Parejas

Producto esperado: Informe escrito con evaluación crítica y propuestas de mejora.

Duración estimada: 3 horas

Actividad 4: Presentación final integradora de análisis y conclusiones

Objetivo: Comunicar de manera clara y estructurada el análisis y conclusiones derivadas del estudio de casos.

Descripción:

- Cada estudiante prepara una presentación oral individual basada en los análisis realizados en actividades anteriores.
- La presentación debe incluir descripción de la metodología, análisis comparativo, evaluación crítica, lecciones aprendidas y recomendaciones.
- Se promueve el uso de ayudas visuales (diapositivas, gráficos, tablas).
- Se realiza una sesión de preguntas y respuestas para fomentar la argumentación.

Organización: Individual

Producto esperado: Presentación oral con soporte visual y defensa de conclusiones.

Duración estimada: 1 hora por estudiante (dependiendo del tamaño del grupo, podría dividirse en varias sesiones)

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre metodologías de desarrollo y capacidad inicial para analizar casos.

Cómo se evalúa: Cuestionario corto o encuesta escrita con preguntas sobre metodologías y análisis básico de casos.

Instrumento sugerido: Test de opción múltiple y preguntas abiertas breves.

Evaluación formativa

Qué se evalúa: Progreso en análisis de casos, comparación de metodologías y capacidad crítica.

Cómo se evalúa: Revisión continua de informes parciales, participación en debates y retroalimentación durante actividades grupales y en parejas.

Instrumento sugerido: Rúbricas para evaluación de informes escritos y participación, listas de cotejo para presentaciones orales preliminares.

Evaluación sumativa

Qué se evalúa: Dominio integral para analizar, comparar, evaluar críticamente y comunicar conclusiones sobre casos reales.

Cómo se evalúa: Evaluación final basada en la presentación oral integradora y el informe escrito completo que incluye análisis, comparación, evaluación crítica y recomendaciones.

Instrumento sugerido: Rúbrica detallada que mida claridad, profundidad analítica, fundamentación de argumentos, calidad de propuestas y habilidades comunicativas.

Unidad 15: Taller Práctico de Aplicación de Metodologías

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de seleccionar y justificar la metodología de desarrollo de software más adecuada para un proyecto específico en equipo, considerando las características y requerimientos del mismo.
- Al finalizar la unidad, el estudiante será capaz de aplicar de manera práctica las fases y actividades de la metodología seleccionada en el desarrollo colaborativo de un proyecto de software, siguiendo las mejores prácticas aprendidas.
- Al finalizar la unidad, el estudiante será capaz de documentar y comunicar los avances y resultados del proyecto utilizando herramientas y técnicas de gestión de proyectos, asegurando la claridad y coherencia en la presentación.
- Al finalizar la unidad, el estudiante será capaz de evaluar el proceso y los resultados obtenidos durante el desarrollo del proyecto, proponiendo mejoras basadas en criterios de calidad y eficiencia.

Contenidos Temáticos

1. Introducción al Taller Práctico

- Objetivos y alcance del taller: explicación del propósito de aplicar metodologías en un contexto real.
- Formación de equipos y asignación de roles: definición de roles clave (líder, analista, desarrollador, tester, etc.) para un trabajo colaborativo efectivo.
- Selección del proyecto de software: criterios para elegir un proyecto adecuado a los objetivos y alcance del taller.

2. Selección y Justificación de la Metodología de Desarrollo

- Revisión de metodologías estudiadas: Agile, Scrum, Kanban, Waterfall, DevOps, etc.
- Análisis de características del proyecto: tamaño, complejidad, requerimientos, equipo, plazos.
- Criterios para la selección de la metodología adecuada: alineación con requerimientos y contexto del proyecto.
- Elaboración de la justificación de la metodología elegida: argumentos técnicos y organizacionales.

3. Planificación y Organización del Proyecto

- Definición de alcance y entregables del proyecto.
- Planificación de fases y actividades según la metodología seleccionada.
- Herramientas de gestión de proyectos: uso de tableros, cronogramas, gestión de tareas y comunicación.
- Asignación de responsabilidades y establecimiento de canales de comunicación.

4. Desarrollo Colaborativo del Proyecto Aplicando la Metodología

- Implementación práctica de las fases de la metodología: análisis, diseño, desarrollo, pruebas, despliegue.
- Aplicación de mejores prácticas: integración continua, revisiones, reuniones de seguimiento.
- Gestión de cambios y ajustes durante el desarrollo.
- Documentación continua del proceso y resultados parciales.

5. Documentación y Comunicación de Avances y Resultados

- Elaboración de reportes de avance: contenido, estructura y frecuencia.
- Uso de herramientas colaborativas para la documentación y presentación.
- Comunicación efectiva entre equipo y con stakeholders.
- Preparación y presentación de entregables finales.

6. Evaluación del Proceso y Resultados

- Definición de criterios de calidad y eficiencia para la evaluación.
- Instrumentos para la evaluación del proceso: autoevaluación, coevaluación, revisión por pares.
- Análisis de resultados obtenidos: cumplimiento de objetivos, calidad del software, eficiencia en el desarrollo.
- Propuestas de mejora continua basadas en la evaluación.

Actividades

Actividad 1: Selección y Justificación de la Metodología

Objetivo: Desarrollar la capacidad para seleccionar y justificar la metodología de desarrollo más adecuada para un proyecto específico.

Descripción:

- En equipos, se presenta un caso de proyecto con características y requerimientos específicos.

- Analizan las metodologías aprendidas y comparan cuál se ajusta mejor al caso.
- Elaboran un documento escrito con la metodología seleccionada y una justificación detallada basada en criterios técnicos y organizacionales.
- Presentan su elección y justificación ante el grupo para retroalimentación.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Documento justificado de selección metodológica y presentación oral.

Duración estimada: 3 horas.

Actividad 2: Planificación del Proyecto con la Metodología Seleccionada

Objetivo: Aplicar la planificación detallada de las fases y actividades del proyecto según la metodología elegida.

Descripción:

- En equipos, desarrollan un plan de proyecto que incluya fases, actividades, cronograma y asignación de roles.
- Utilizan herramientas digitales (p.ej., Trello, Jira, MS Project) para organizar y visualizar la planificación.
- Realizan una reunión simulada de planificación para ajustar detalles y definir mecanismos de seguimiento.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Plan de proyecto digital y acta de reunión de planificación.

Duración estimada: 4 horas.

Actividad 3: Desarrollo y Gestión Colaborativa del Proyecto

Objetivo: Aplicar de manera práctica las fases y actividades de la metodología en el desarrollo colaborativo del proyecto.

Descripción:

- Durante varias sesiones, el equipo realiza el desarrollo del proyecto siguiendo el plan y metodología seleccionada.
- Documentan avances, registran incidencias y gestionan cambios mediante las herramientas definidas.
- Implementan prácticas como reuniones diarias (stand-ups), revisiones de código y pruebas continuas, según corresponda.
- Preparan reportes periódicos de avance para compartir con el docente y compañeros.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Código fuente, documentación de avance, reportes periódicos y actas de reuniones.

Duración estimada: 15-20 horas distribuidas en varias sesiones.

Actividad 4: Evaluación y Propuesta de Mejora del Proceso

Objetivo: Evaluar el proceso y resultados del proyecto y proponer mejoras basadas en criterios de calidad y eficiencia.

Descripción:

- Cada equipo realiza una autoevaluación y coevaluación del proceso de desarrollo y resultados obtenidos.

- Analizan fortalezas, debilidades, oportunidades de mejora y amenazas (FODA) del proyecto y la metodología aplicada.
- Elaboran un informe de evaluación que incluya propuestas concretas de mejora para futuros proyectos.
- Presentan el informe y reciben retroalimentación del docente y compañeros.

Organización: Grupos de 4-5 estudiantes.

Producto esperado: Informe de evaluación y presentación de propuestas de mejora.

Duración estimada: 3 horas.

Evaluación

Evaluación Diagnóstica

Qué se evalúa: Conocimientos previos sobre metodologías de desarrollo de software y habilidades para análisis de proyectos.

Cómo se evalúa: Cuestionario corto con preguntas de selección múltiple y análisis de casos breves.

Instrumento sugerido: Prueba en línea o en papel con 15 preguntas y un breve caso para análisis.

Evaluación Formativa

Qué se evalúa: Progreso en la selección, planificación, desarrollo y documentación del proyecto; participación y trabajo en equipo.

Cómo se evalúa: Revisión continua de entregables parciales, observación de dinámicas de equipo, retroalimentación en presentaciones y reportes.

Instrumento sugerido: Rúbricas para evaluación de documentos y presentaciones, listas de cotejo para seguimiento de actividades y participación.

Evaluación Sumativa

Qué se evalúa: Calidad y coherencia en la selección y justificación de la metodología, aplicación práctica en el desarrollo del proyecto, calidad de la documentación y comunicación, y profundidad de la evaluación final y propuestas de mejora.

Cómo se evalúa: Calificación de entregables finales: informe de justificación, plan de proyecto, producto desarrollado, reportes de avance y evaluación final.

Instrumento sugerido: Rúbrica integral que contemple aspectos técnicos, organizacionales, de comunicación y autoevaluación/co-evaluación.

Unidad 16: Presentación y Evaluación Final

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de presentar de forma clara y estructurada el proyecto final, integrando los conceptos y metodologías aprendidas durante el curso, utilizando recursos visuales adecuados.
- Al finalizar la unidad, el estudiante será capaz de analizar y sintetizar la retroalimentación recibida durante la presentación para identificar fortalezas y áreas de mejora en el proyecto.
- Al finalizar la unidad, el estudiante será capaz de evaluar de manera crítica el proceso y resultado del proyecto final, aplicando criterios de calidad y mejores prácticas en desarrollo de software.
- Al finalizar la unidad, el estudiante será capaz de comunicar de forma efectiva los resultados y aprendizajes obtenidos en el proyecto final, adaptando el discurso a diferentes audiencias académicas y profesionales.

Contenidos Temáticos

1. Preparación de la Presentación del Proyecto Final

- 1.1 Estructura de una presentación efectiva: introducción, desarrollo y conclusión
- 1.2 Integración de conceptos y metodologías del curso en la presentación
- 1.3 Uso adecuado de recursos visuales: diapositivas, diagramas, demos y prototipos
- 1.4 Técnicas de comunicación oral y lenguaje corporal para presentaciones académicas y profesionales

2. Dinámica de Presentación y Recepción de Retroalimentación

- 2.1 Estrategias para presentar con claridad y confianza frente a diferentes audiencias
- 2.2 Métodos para la recolección de retroalimentación: preguntas, encuestas y comentarios
- 2.3 Análisis crítico de la retroalimentación: identificación de fortalezas y áreas de mejora
- 2.4 Técnicas para responder preguntas y manejar críticas constructivamente

3. Evaluación Crítica del Proyecto Final

- 3.1 Criterios de calidad en proyectos de desarrollo de software: funcionalidad, usabilidad, mantenibilidad y rendimiento
- 3.2 Aplicación de mejores prácticas y estándares en el desarrollo y documentación del proyecto
- 3.3 Autoevaluación y coevaluación: herramientas y métodos
- 3.4 Reflexión sobre el proceso de desarrollo y aprendizajes obtenidos

4. Comunicación de Resultados y Aprendizajes

- 4.1 Adaptación del discurso a audiencias académicas y profesionales
- 4.2 Elaboración de informes y resúmenes ejecutivos claros y concisos
- 4.3 Uso de lenguaje técnico y no técnico según el público
- 4.4 Estrategias para destacar contribuciones personales y del equipo

Actividades

Actividad 1: Diseño y Ensayo de la Presentación Final

Objetivo: Presentar de forma clara y estructurada el proyecto final, integrando conceptos y metodologías del curso.

Descripción:

- Los estudiantes preparan una presentación de 10-15 minutos sobre su proyecto final, siguiendo una estructura predefinida.
- Incluyen diapositivas con recursos visuales que reflejen las metodologías y conceptos aprendidos.
- Realizan un ensayo de la presentación frente a un grupo reducido para recibir retroalimentación inicial.
- Incorporan mejoras con base en la retroalimentación recibida.

Organización: Individual o en parejas

Producto esperado: Presentación final en formato digital (diapositivas) y ensayo grabado o realizado en clase.

Duración estimada: 3 horas

Actividad 2: Simulación de Presentación con Retroalimentación

Objetivo: Analizar y sintetizar la retroalimentación recibida para identificar fortalezas y áreas de mejora.

Descripción:

- Los estudiantes presentan su proyecto final ante la clase, simulando un entorno profesional o académico.
- Los compañeros y el docente ofrecen retroalimentación estructurada usando una rúbrica específica.
- Cada estudiante o equipo recoge la retroalimentación y realiza un análisis crítico para identificar fortalezas y oportunidades de mejora.

Organización: Grupos pequeños para presentación y retroalimentación, análisis individual

Producto esperado: Documento de análisis de retroalimentación y plan de mejora.

Duración estimada: 4 horas

Actividad 3: Evaluación Crítica y Reflexión del Proyecto

Objetivo: Evaluar críticamente el proceso y resultado del proyecto aplicando criterios de calidad y mejores prácticas.

Descripción:

- Los estudiantes elaboran un informe donde aplican criterios de calidad para evaluar su proyecto final.
- Incluyen una reflexión sobre el proceso, destacando lo que funcionó bien y lo que se podría mejorar.
- Comparan su proyecto con estándares o casos de referencia para fundamentar su evaluación.

Organización: Individual

Producto esperado: Informe escrito de evaluación crítica y reflexión.

Duración estimada: 2 horas

Actividad 4: Comunicación Adaptada de Resultados y Aprendizajes

Objetivo: Comunicar efectivamente resultados y aprendizajes adaptando el discurso a diferentes audiencias.

Descripción:

- Los estudiantes preparan dos versiones de un resumen ejecutivo del proyecto: una para una audiencia técnica (profesores y especialistas) y otra para una audiencia no técnica (usuarios o clientes).
- Presentan ambos resúmenes ante compañeros que simulan las audiencias objetivo y reciben retroalimentación sobre claridad y adecuación del lenguaje.

Organización: Individual o parejas

Producto esperado: Dos resúmenes ejecutivos escritos y presentación oral breve de ambos.

Duración estimada: 3 horas

Evaluación**Evaluación Diagnóstica**

Qué se evalúa: Conocimientos previos sobre presentación de proyectos y manejo de retroalimentación.

Cómo se evalúa: Encuesta o cuestionario breve al inicio de la unidad con preguntas sobre experiencia en presentaciones, uso de recursos visuales y recepción de feedback.

Instrumento sugerido: Cuestionario en línea o en papel con preguntas cerradas y abiertas.

Evaluación Formativa

Qué se evalúa: Progreso en la preparación y ensayo de la presentación, capacidad para integrar retroalimentación y análisis crítico.

Cómo se evalúa: Observación directa de ensayos, revisión de borradores de presentaciones y análisis de retroalimentación, aportes durante sesiones de retroalimentación.

Instrumento sugerido: Lista de cotejo para presentaciones, rúbrica para análisis de retroalimentación, retroalimentación cualitativa del docente y compañeros.

Evaluación Sumativa

Qué se evalúa: Presentación final del proyecto, análisis crítico del proyecto, y comunicación adaptada de resultados y aprendizajes.

Cómo se evalúa: Evaluación mediante rúbricas que consideren claridad, integración de contenidos, calidad de recursos visuales, análisis de retroalimentación, aplicación de criterios de calidad, y adecuación comunicativa.

Instrumento sugerido: Rúbrica detallada para presentación oral y escrita, informe de evaluación crítica, y resúmenes ejecutivos.