



Proceso de desarrollo de software

+ Clase	�편집기 <u>Entornos de Desarrollo</u>
StateManager Estado	Completado
StateManager Tipo	Entrega
StateManager Fecha de Entrega	@23 de octubre de 2025
StateManager Fecha de Finalización	@20 de octubre de 2025 16:07

ACTIVIDAD

Enunciado del Ejercicio

Crea una tabla comparativa de las ventajas e inconvenientes de cada uno de los modelos de desarrollo de software analizados en este capítulo:

Cascada, prototipado, incremental, espiral, scrum, kanban y programación extrema.

Tabla Comparativa de Modelos de Desarrollo

Modelo	Ventajas	Inconvenientes
Cascada	<ul style="list-style-type: none"> • Simple y ordenado. • Bueno para proyectos con requisitos fijos. 	<ul style="list-style-type: none"> • Muy rígido, no permite cambios. • Los errores se detectan tarde y son caros.
Prototipado	<ul style="list-style-type: none"> • El cliente opina desde el principio. • Reduce el riesgo de que el producto final no guste. 	<ul style="list-style-type: none"> • Puede generar falsas expectativas. • A veces se descuida la calidad por ir rápido.
Incremental	<ul style="list-style-type: none"> • Se entregan partes funcionales del software pronto. • Es flexible y permite hacer ajustes. 	<ul style="list-style-type: none"> • Necesita una buena planificación inicial. • El coste total puede ser más alto.
Espiral	<ul style="list-style-type: none"> • Se centra en gestionar y reducir riesgos. • Ideal para proyectos muy grandes y complejos. 	<ul style="list-style-type: none"> • Es un modelo muy complejo y caro. • No es adecuado para proyectos pequeños.
Scrum	<ul style="list-style-type: none"> • Muy adaptable a los cambios. • Entregas rápidas y frecuentes de software funcional. 	<ul style="list-style-type: none"> • Es difícil fijar un presupuesto y fecha final. • Requiere un equipo muy disciplinado.
Kanban	<ul style="list-style-type: none"> • Máxima flexibilidad para cambiar prioridades. • Mejora el flujo de trabajo de forma continua. 	<ul style="list-style-type: none"> • La falta de plazos fijos puede llevar a retrasos. • Necesita que el equipo sea muy constante.
XP	<ul style="list-style-type: none"> • Se enfoca en la alta calidad del código. • Muy adaptable y colaborativo. 	<ul style="list-style-type: none"> • Requiere mucha disciplina técnica. • Puede ser difícil de implementar en algunas empresas.