

Ejercicio 1: Investiga los lenguajes de programación que se han mencionado y detalla en la siguiente tabla sus características:

	Nivel de abstracción	Propósito	¿Compilado o interpretado?	¿Imperativo o declarativo?	Paradigma	Otros
Python	abstracción mayor	ciencia de datos, desarrollo web, automatización	interpretado	imperativo	multiparadigma	
Java	alta	lenguaje multiplataforma	compilado	imperativo	multiparadigma	
C++	medio	control de hardware, POO, eficiencia y optimización	compilado	imperativo	multiparadigma	
Javascript	alto	desarrollo web, móvil, juegos	interpretado	imperativo	multiparadigma	
C#	alto	aplicaciones de escritorio	compilado	imperativo	multiparadigma	
R	alto	análisis estadístico, ciencia de datos	interpretado	declarativo	multiparadigma	
PHP	alto	desarrollo web	interpretado	imperativo	multiparadigma	
Kotlin	alto	aplicaciones android	compilado	imperativo	multiparadigma	

Swift	alto	aplicaciones para apple	compilado	imperativo	multiparadigma	
Go	medio alto	aplicaciones de servidor, microservicios, herramientas de red,	compilado	imperativo	procedural	

Ejercicio 2: Razona las ventajas/desventajas de los compiladores e intérpretes.

Compiladores		Intérpretes	
Ventajas	rendimiento, detección de errores, código independiente	Ventajas	desarrollo eficaz, flexibilidad, portabilidad
Inconvenientes	tiempo de compilación, flexibilidad baja, dependencia	Inconvenientes	bajo rendimiento, errores a la hora de la ejecución, depende del entorno

Ejercicio 3: Investiga sobre los siguientes ciclos de vida: Modelo Incremental, Modelo en Espiral.

Modelo Incremental		Modelo en Espiral	
Fases	planificación, desarrollo, prueba, implementación	Fases	planificación, análisis de riesgo, desarrollo, evaluación repetición

Ventajas	flexibilidad riesgo menor, una entrega rápida	Ventajas	menor gestión de riesgo, gran adaptabilidad, enfoque hacia el cliente
Inconvenientes	una difícil planificación, depende de la arquitectura, una integración compleja	Inconvenientes	gran complejidad, gran coste, requiere una buena documentación
Sistemas	aplicaciones web sistemas de gestión	Sistemas	proyectos grandes, sistemas embebidos

sistema embebido

Es un sistema de computación diseñado para realizar tareas específicas dentro de un dispositivo más grande. A menudo, estos sistemas están integrados en hardware y no son directamente accesibles por el usuario, lo que los diferencia de las computadoras de propósito general.

Características de los sistemas embebidos: Funcionalidad específica, Recursos limitados, memoria y almacenamiento, Integración de hardware y software, Tiempo real, Bajo consumo de energía

Ejemplos: Electrodomésticos, Automóviles, Dispositivos médicos.