INFORME COMPARATIVO DE LOS ETILOS DE PROGRAMACION

Estilo de Programación	Descripción	Ventajas	Desventajas	Lenguajes
Programación Imperativa	Está relacionada con el uso instrucciones que cambian el estado del programa mediante la manipulación directa de variables y la ejecución de comandos secuenciales.	Fácil de entender y aprender.	Difícil de mantener en programas grandes.	C, Python (imperativo), JavaScript.
Programación Declarativa	Este esta enfocado en lo que se quiere lograr sin especificar el como hacerlo	Menos código y mayor claridad en algunos contextos.	Menos control sobre el rendimiento y la implementación.	SQL, HTML, CSS
Programación Funcional	Se encarga de la evaluación de funciones matemáticas, evita el cambio de estado y los efectos secundarios.	Facilita la concurrencia y el razonamiento matemático sobre el código.	Menos intuitivo para quienes están acostumbrados a otros estilos.	Haskell. Lisp, Scala.
Programación Orientada a Objetos	Organiza el código en objetos los cuales combinan datos y comportamientos.	Modularidad y reutilización de código.	Puede volverse complejo y sobrecargado.	Java, C++, Python
Programación Procedural	Similar al imperativo, pero más enfocado en procedimientos o rutinas reutilizables.	Promueve la reutilización de código y la estructura clara.	Puede llevar a una estructura rígida y difícil de escalar.	C, Pascal, Fortran
Programación Event-Driven	El flujo del programa es determinado por eventos como interacciones de usuario o mensajes de otros programas.	Adecuado para aplicaciones interactivas o con interfaces gráficas.	Puede ser difícil de depurar debido al flujo de control no lineal.	JavaScript, C#, VB.NET
Programación Lógica	Basado en reglas lógicas y en cómo se relacionan entre sí, en lugar de secuencias de pasos.	Útil para problemas complejos que pueden expresarse mediante reglas y hechos.	A veces es difícil de entender y menos eficiente en términos de ejecución.	Prolog, Datalog
Programación Concurrente	Permite la ejecución de múltiples procesos simultáneamente, facilitando la gestión de tareas en paralelo.	Aprovecha mejor los recursos del sistema y mejora el rendimiento en tareas paralelas.	Manejar la sincronización y la comunicación entre procesos puede ser complejo y propenso a errores.	Go, Erlang, Java
Programación Orientada a Aspectos	Separa las preocupaciones transversales del código principal, como el manejo de errores o la seguridad.	Facilita la modularidad y reduce la duplicación de código.	Puede hacer que el código sea más difícil de seguir y entender.	AspectJ, PostSharp
Programación Basada en Componentes	Descompone el sistema en componentes reutilizables y encapsulados, promoviendo la modularidad.	Facilita el mantenimiento y la evolución del software.	Puede requerir más esfuerzo inicial en diseño y planificación.	Java (Beans), .NET, Web Components

Alumna: Sandy Judith Hernández Carlos