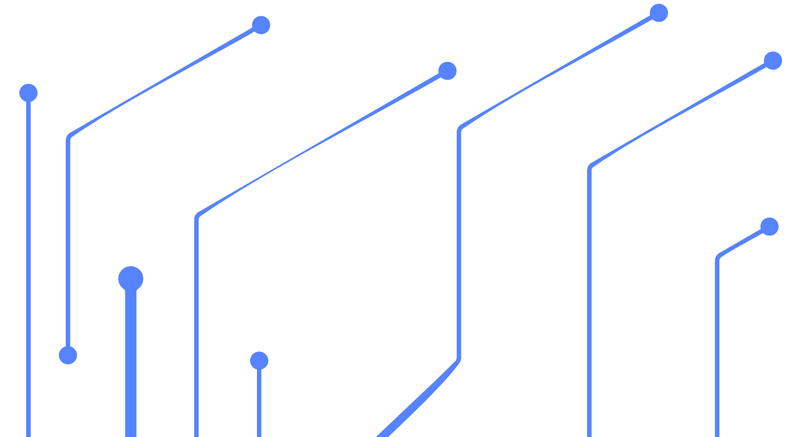
Paradigma Funcional

APLICADO EN C++

Entendiendo los Paradigmas de Programación

LOS PARADIGMAS DE PROGRAMACIÓN SON ENFOQUES FUNDAMENTALES QUE GUÍAN LA MANERA EN QUE LOS DESARROLLADORES ESCRIBEN Y ORGANIZAN EL CÓDIGO DE UN PROGRAMA. ESTOS PARADIGMAS OFRECEN UN CONJUNTO DE REGLAS, PRINCIPIOS Y ESTILOS DE PROGRAMACIÓN QUE DETERMINAN CÓMO SE ABORDAN LOS PROBLEMAS Y SE CONSTRUYEN LAS SOLUCIONES.



Beneficios de Adoptar un Paradigma de Programación

Una de las ventajas es la reutilización de código, que permite a los programadores aprovechar soluciones existentes y componentes de software para construir nuevos sistemas. Al seguir los principios de un paradigma específico, como el paradigma orientado a objetos, los desarrolladores pueden encapsular funcionalidades en clases y objetos reutilizables, reduciendo así el tiempo y el esfuerzo requeridos para escribir código desde cero.

Desafíos y Limitaciones de los Paradigmas de Programación

A pesar de las ventajas que ofrecen, los paradigmas de programación también pueden presentar ciertos desafíos y limitaciones que los desarrolladores deben considerar.

CONCEPTOS ABSTRACTOS

El paradigma funcional se basa en conceptos abstractos como funciones puras, inmutabilidad y recursión. Para los programadores acostumbrados a paradigmas imperativos o orientados a objetos, estos conceptos pueden resultar difíciles de entender y aplicar correctamente, lo que puede aumentar la curva de aprendizaje y la complejidad del código.

MANIPULACION DE ESTADOS

El enfoque en funciones puras y datos inmutables puede limitar la capacidad de manipular estados y realizar efectos secundarios en un programa. Si bien esto puede conducir a un código más predecible y fácil de razonar, también puede hacer que sea más difícil interactuar con sistemas externos, como bases de datos o interfaces de usuario, que dependen de la modificación de estados.

RENDIMIENTO Y EFICIENCIA

Algunos aspectos del paradigma funcional, como la recursión y el uso intensivo de funciones de orden superior, pueden afectar el rendimiento y la eficiencia del programa, especialmente en situaciones donde se requiere un procesamiento intensivo de datos. La optimización de código funcional puede ser más complicada que en otros paradigmas, lo que puede llevar a aplicaciones menos eficientes en términos de velocidad y uso de recursos.

MENOR ADOPCION EN LA INDUSTRIA

Aunque el paradigma funcional ha ganado popularidad en los últimos años, especialmente con el aumento de los lenguajes de programación funcionales como Haskell, Scala y Clojure, todavía tiene una adopción más limitada en comparación con paradigmas más establecidos como el orientado a objetos. Esto puede dificultar la integración de habilidades y herramientas funcionales en equipos de desarrollo y proyectos existentes, y limitar las oportunidades profesionales para los programadores especializados en este paradigma.

Lenguajes de Programación que Soportan el Paradigma

FACEBOOK NVIDIA MICROSOFT

Haskell

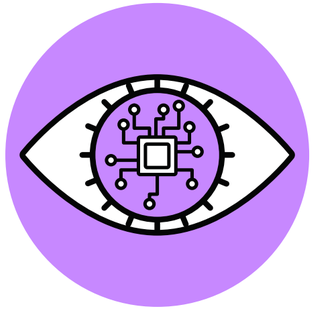
TWITTER LINKEDID NETFLIX

Scala

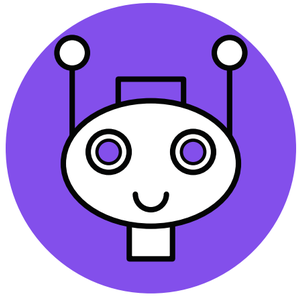
WALMART AMAZON WEB SERVICE (AWS

Cloujure

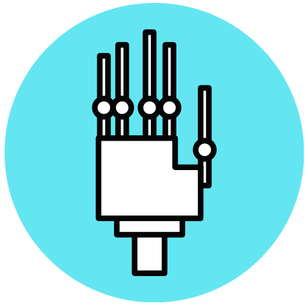
Especialidad



PROCESAMIENTO DE DATOS



DESARROLLO DE SISTEMAS DE INTELIGENCIA ARTIFICIAL Y APRENDIZAJE AUTOMÁTICO



ANÁLISIS MATEMÁTICO Y CIENTÍFICO

PROGRAMACIÓN CONCURRENTE Y PARALELA

Comunidades

STACK OVERFLOW

REDDIT
(R/PROGRAMMING)

GITHUB

LAMBDA
ISLAND