

Sistema Integral de Evaluación para Lenguajes de Programación (SIE-LP)

Introducción:

Asd

Niveles:

El modelo se divide en 3 niveles de conocimiento del lenguaje:

- Short
- Medium
- Long

Niveles de conocimiento del Lenguajes de Programación		
SHORT	S-1	<p>Sintaxis: Entiende, reconoce y diferencia un conjunto limitado y de uso común de palabras reservadas dentro del lenguaje de programación.</p> <p>Manejo de Datos: Posee un conocimiento limitado de uno o mas tipos de datos que se utilizan dentro del lenguaje de programación.</p> <p>Operaciones: Es capaz de realizar operaciones aritméticas, lógicas, de asignación, comparación, concatenación o conversión de uno o mas tipos de datos.</p> <p>Uso de estructuras: Usa y reconoce una o varias estructuras simples de control de flujo y loops.</p> <p>Manejo de errores: Reconoce y resuelve errores simples de tipo lógico o de sintaxis.</p> <p>Documentación: Reconoce y utiliza la estructura básica de comentarios.</p> <p>Consulta: Necesita y realiza consulta activa de documentación, tutoriales, IA o un asesor para entender y crear código simple.</p> <p>Codificación: Es capaz de realizar programas simples que incluyan operaciones aritméticas simples, mensajes por consola y usa estructuras de control de flujo de código básicas</p>

	S-2	<p>Sintaxis: Entiende, reconoce y diferencia un conjunto básico y común de palabras reservadas dentro del lenguaje de programación.</p> <p>Manejo de Datos: Conoce y entiende las ventajas, desventajas y aplicaciones de los diferentes tipos de datos que se utilizan dentro del lenguaje de programación.</p> <p>Operaciones: Es capaz de realizar operaciones aritméticas, lógicas, de asignación, comparación, concatenación o conversión de de forma común y sin necesidad de consulta externa.</p> <p>Uso de estructuras: Usa y reconoce las ventajas y desventajas de las estructuras de control de flujo y loops en el lenguaje de programación.</p> <p>Manejo de errores: Reconoce, resuelve, predice y utiliza estructuras simples que evitan errores en tipos de datos y de lógica en el código.</p> <p>Documentación: Reconoce la necesidad de ar nombres descriptivos a las variables, ademas de hacer uso básico de la sintaxis para crear comentarios útiles.</p> <p>Consulta: Aun hace uso de la IA, documentación, tutoriales o un asesor, revisar estructuras de código o identificar errores, pero es menos frecuente y puede hacer uso de estructuras comunes sin necesidad de la consulta.</p> <p>Codificación: Es capaz de realizar programas que permiten la interacción entre tipos de datos y el uso de entradas y salidas, es capaz de resolver algoritmos simples y de usarlos en su programa consultando material externo.</p>
MEDIUM	M-1	<p>Sintaxis: Posee un cocimiento extenso de la sintaxis del lenguaje ademas de que es capaz de reconocerlo y diferenciar la sintaxis de este de la sintaxis de otros lenguajes de programación.</p> <p>Manejo de Datos: Conoce los diferentes tipos de datos que se utilizan dentro del lenguaje de programación, ademas de conocer estructuras de agrupación de datos como arrays, mapas, objetos, clases etc.</p> <p>Operaciones: Es capaz de usar, entender y crear operaciones complejas que afecten a datos o a estructuras de datos, como son algoritmos de búsqueda, reemplazo, identificacion, agrupación, etc.</p> <p>Uso de estructuras: Crea y usa bloques de código re-utilizable como son, funciones, clases, etc.</p> <p>Manejo de errores: Hace uso y crea estructuras diseñadas para prevenir errores de código, ademas de conocer y usar de forma simple los métodos de manejo de errores del lenguaje.</p>

		<p>Documentación: Hace el uso de las “buenas practicas” del lenguaje, comentarios simples y útiles, además de agrupar de usar una estructura lógica en su código.</p> <p>Consulta: Hace uso de la IA, documentación, tutoriales o un asesor, para entender estructuras complejas o para identificar y resolver errores de código no tan comunes. También entiende la importancia de buscar en varias fuentes y comparar ideas.</p> <p>Codificación: Es capaz de crear programas simples pero que cuentan con múltiples funcionalidades, además de que es capaz de hacer uso de funciones, objetos y librerías.</p>
	M-2	<p>Sintaxis: Lee, comprende e identifica patrones básicos de diseño, nomenclatura, estilo de código, regularización, el uso de códigos re-utilizables en su propio código o en el de alguien más.</p> <p>Manejo de Datos: Usa de forma eficiente los diferentes tipos de datos y estructuras de datos usando modularidad que mejora la lectura y limpieza del código.</p> <p>Operaciones: Utiliza, adapta o crea algoritmos y/o soluciones específicas para un problema.</p> <p>Uso de estructuras: Se enfoca en la modularidad y la reutilización del código creando librerías y funciones con parámetros de retorno.</p> <p>Manejo de errores: Conoce y hace uso activo de funciones para minimizar los errores en el código, además de prevenirlos y permitir su correcta manipulación con estructuras “try-catch”.</p> <p>Documentación: Crea una documentación de referencia para el proyecto, agrega comentario cortos, simples y precisos que ayuden a entender como funciona cada parte.</p> <p>Consulta: Usa la documentación, IA o colegas como ayuda para identificar y resolver excepciones del lenguaje además de buscar formas para optimizar el código.</p> <p>Codificación: Es capaz de crear programas complejos o librerías con múltiples funcionalidades, que sean accesibles, adaptables y fáciles de usar para otros programadores.</p>
EXTEND	L-1	<p>Sintaxis: Lee, comprende e identifica patrones de diseño, nomenclatura, estilo de código, regularización, el uso de códigos re-utilizables en su propio código o en el de alguien más, además de encontrar soluciones y mejoras a estos sin problema.</p> <p>Manejo de Datos: Es capaz de identificar problemas de uso y mejoras de optimización en el manejo de estructuras de datos.</p> <p>Operaciones:</p>

	<p>Crea y optimiza funciones o programas que realizen operaciones con datos, mejorando al eficiencia en tiempo y uso de memoria.</p> <p>Uso de estructuras: Crea y usa bloques de código eficientes y re-utilizables, aplicando el correcto nivel de modularidad a todo un proyecto.</p> <p>Manejo de errores: Crea y usa estructuras de manejo de errores de forma eficiente, ademas de incluir mensajes útiles que ayuden a otros a entender y corregir el error.</p> <p>Documentación: Conoce y hace uso de las buenas practicas en su código sin esfuerzo, ademas de llevar una buena documentación de sus proyectos para que otros puedan colaborar en ellos.</p> <p>Consulta: Hace uso de la documentación, colegas o la IA, para encontrar mejoras en el código para optimizar procesos, tiempo de ejecución y uso de memoria,</p> <p>Codificación: Es capaz de crear programas complejos con buen nivel de modularidad y multi-funcionalidad, que ademas de estar bien implementados hacen un buen uso de la memoria y la velocidad de ejecucion.</p>
L-2	<p>Sintaxis: Lee, comprende e identifica patrones de diseño, nomenclatura, estilo de código, regularización, el uso de códigos re-utilizables en su propio código o en el de alguien mas, ademas de encontrar soluciones y mejoras a estos sin problema.</p> <p>Manejo de Datos: Es capaz de identificar problemas de uso y mejoras de optimización en el manejo de estructuras de datos.</p> <p>Operaciones: Crea y optimiza funciones o programas que realizen operaciones con datos, mejorando al eficiencia en tiempo y uso de memoria.</p> <p>Uso de estructuras: Crea y usa bloques de código eficientes y re-utilizables, aplicando el correcto nivel de modularidad a todo un proyecto.</p> <p>Manejo de errores: Crea y usa estructuras de manejo de errores de forma eficiente, ademas de incluir mensajes útiles que ayuden a otros a entender y corregir el error.</p> <p>Documentación: Conoce y hace uso de las buenas practicas en su código sin esfuerzo, ademas de llevar una buena documentación de sus proyectos para que otros puedan colaborar en ellos.</p> <p>Consulta: Hace uso de la documentación, colegas o la IA, para encontrar mejoras en el código para optimizar procesos, tiempo de ejecución y uso de memoria,</p> <p>Codificación: Es capaz de crear programas complejos con múltiples funcionalidades, ademas de que es capaz de hacer uso de funciones, objetos y librerías.</p>

Cuadro extendido de habilidades

SHORT	S-1	<ul style="list-style-type: none"> Tiene conocimiento basico de la sintaxis del lenguaje Conoce palabras basicas Tiene conocimientos de los tipos basicos de datos sabe como usar espacios de memoria para guardar datos simples conoce al menos un sistema de control de flujo conoce y usa almenos un sistema para crear bulces comprende la sintaxis basica y reconoce estructuras simples puede realizar operacones con los tipos de datos que conoce conoce reconoce y resuelve errore de sintaxis en el codigo. puede realizar programas que expresen salida por consola o que realicen operaciones matematicos, logicas o de control, simples. Puede realizar estructuras combinadas e logica, loops, exreciones matematicas o logicas con ayuda de una IA, un asesor mas avanzado o a partir de material de consula.
	S-2	<ul style="list-style-type: none"> Tiene concimiento de diversos tipos de datos en el lenguaje ademas de poder realizar operaciones entre ellos. Tiene nocion del uso de estructuras de control de flujo dentro del lenguaje de programacion, sabe concatenar y crear ecepciones. Conoce estructuras de Loop ademas de las ventajas y desventajas del uso de cada una. Reconoce la estructura basica de la sintaxis del lenguaje reconoce las ventajas deluso de la memoria, crea nombres de variables reconocibles. Tiene conociemitno inicial en el uso de etructuras de memoriamaas compleja como arrays
LongMEDIUM	M-1	
	M-2	
	L-1	
	L-2	