

Paradigmas de Programación

Alumno: Lázaro León Sánchez

Ejercicio 1

Parte A

Considera el lenguaje JavaScript acotado al paradigma de programación estructurada y analízalo en términos de los cuatro componentes de un paradigma mencionados por Kuhn.

1. Generalización simbólica: ¿Cuáles son las reglas escritas del lenguaje?
2. Creencias de los profesionales: ¿Qué características particulares del lenguaje se cree que sean "mejores" que en otros lenguajes?

- 1) En JavaScript, la programación estructurada implica el uso de secuencias, selección (condicionales como if, else) y repetición (for, while). Aunque JavaScript soporta paradigmas más complejos como el orientado a objetos o funcional, al usar solo programación estructurada, se limitan los elementos a estructuras de control básicas, sin clases o prototipos.

La declaración de variables sigue una estructura flexible con var, let y const, aunque el uso de let y const es preferido para respetar el alcance local de las variables.

Las funciones, esenciales en programación estructurada, se definen con la palabra clave function y pueden almacenarse en variables para un manejo dinámico.

- 2) Para la programación estructurada, muchos consideran que JavaScript es "mejor" por su flexibilidad y simplicidad en la declaración de funciones y variables, lo cual permite desarrollos rápidos y un aprendizaje accesible. La interoperabilidad de JavaScript con el navegador también se percibe como una ventaja, ya que permite construir lógica estructurada en el lado del cliente sin requerir herramientas o compiladores adicionales.

Otro aspecto valorado es su tipado dinámico, que, aunque puede ser una fuente de errores en ciertos casos, permite escribir menos código para manejar estructuras básicas, lo cual es visto como una ventaja en aplicaciones más simples.

Parte B

Considera el lenguaje JavaScript acotado al paradigma de programación estructurada y analízalo en términos de los ejes propuestos para la elección de un lenguaje de programación ([¿Cómo elegir un lenguaje?](#)) y responde:

1. ¿Tiene una sintaxis y una semántica bien definida? ¿Existe documentación oficial?
 2. ¿Es posible comprobar el código producido en ese lenguaje?
 3. ¿Es confiable?
 4. ¿Es ortogonal?
 5. ¿Cuáles son sus características de consistencia y uniformidad?
 6. ¿Es extensible? ¿Hay subconjuntos de ese lenguaje?
 7. El código producido, ¿es transportable?
-
- 1) JavaScript tiene una sintaxis y semántica bien definidas, especialmente en la versión ECMAScript, que estandariza el lenguaje. Existen varias fuentes de documentación oficial, siendo MDN Web Docs una de las más completas, junto con la especificación ECMAScript (ECMA-262).
 - 2) Sí, el código de JavaScript puede ser comprobado usando linters como ESLint para detectar errores comunes o posibles problemas de estilo y prácticas. Además, existen herramientas de testing como Jasmine y Mocha para realizar pruebas automatizadas y asegurar el funcionamiento esperado del código estructurado.
 - 3) JavaScript es confiable para muchas aplicaciones de cliente y servidor, aunque es un lenguaje dinámico, lo cual puede introducir errores difíciles de detectar en tiempo de ejecución.
 - 4) JavaScript tiene una ortogonalidad moderada. Muchas de sus construcciones se pueden combinar de forma predecible, pero también tiene excepciones y peculiaridades en ciertas características, como el alcance de variables (var vs let y const) y el uso de funciones como objetos.

- 5) JavaScript es relativamente consistente y uniforme. La mayoría de sus estructuras de control (como if, for, while) siguen un formato común. Sin embargo, algunas inconsistencias en el comportamiento de ciertos operadores y en la conversión de tipos pueden afectar esta uniformidad en entornos más avanzados.
- 6) JavaScript es extensible. Se puede ampliar mediante bibliotecas y frameworks externos, además de poder combinarse con otros lenguajes.
- 7) Sí, el código JavaScript es altamente transportable. Al estar basado en la web y ejecutarse en cualquier navegador moderno, el código puede ejecutarse en diferentes sistemas operativos y dispositivos sin cambios significativos.