**Paradigmas de Programación**

**Aplicando lo Aprendido N°1:**

**Ejercicio N°1**

**Respuestas:**

**Parte A:**

1) Generalización simbólica: ¿Cuáles son las reglas escritas del lenguaje?

En JavaScript estructurado, las reglas escritas del lenguaje incluyen:

* Sintaxis de control de flujo: uso de if, else, switch, for, while, do-while.
* Funciones: definición de bloques de código reutilizables con function nombreFunción(...) { ... }.
* Variables: declaración con var, let o const.
* Tipos de datos básicos: number, string, boolean, null, undefined, object (aunque en programación estructurada se usan poco los objetos).
* Operadores: aritméticos (+, -, \*, /), lógicos (&&, ||, !) y de comparación (==, ===, <, >).
* Entrada/Salida simple: por ejemplo, console.log() para salida y prompt() para entrada en un navegador.

2) Creencias de los profesionales: ¿Qué características particulares del lenguaje se cree que sean "mejores" que en otros lenguajes?

Sobre lo que se considera “mejor” en JavaScript estructurado:

* Simplicidad y rapidez para empezar: muchos programadores creen que JavaScript es fácil de aprender para tareas pequeñas o prototipos.
* Flexibilidad: no requiere declarar tipos estrictos (tipado dinámico), lo que permite escribir código rápido y flexible.
* Compatibilidad con navegadores: es el lenguaje estándar del navegador, lo que lo hace universal para desarrollo web.
* Gran soporte comunitario: hay muchísima documentación y librerías disponibles, aunque en programación estructurada esto no se usa tanto como en frameworks modernos.

**Parte B:**

1) ¿Tiene una sintaxis y una semántica bien definida? ¿Existe documentación oficial?

* Sí, JavaScript tiene una sintaxis y semántica formalmente definida por el estándar ECMAScript, mantenido por ECMA International.
* La documentación oficial más consultada es la de MDN Web Docs, que ofrece descripciones precisas, ejemplos y referencias cruzadas.
* Además, el estándar ECMAScript está disponible públicamente y se actualiza regularmente.

2) ¿Es posible comprobar el código producido en ese lenguaje?

Sí, aunque JavaScript es un lenguaje interpretado, existen herramientas para comprobar el código:

* Linting: Herramientas como ESLint permiten verificar errores de estilo y lógica.
* Testing: Frameworks como Jest, Mocha o Jasmine permiten pruebas unitarias y funcionales.
* Debugging: Navegadores modernos ofrecen consolas y depuradores integrados para inspección paso a paso.

3) ¿Es confiable?

Sí, con condiciones. JavaScript es confiable si se usa con buenas prácticas:

* Su tipado dinámico puede causar errores si no se controla adecuadamente.
* La confiabilidad mejora con el uso de herramientas como TypeScript, que añade tipado estático.
* En el paradigma estructurado, la claridad del flujo secuencial y modular ayuda a mantener la confiabilidad.

4) ¿Es ortogonal?

Parcialmente. JavaScript tiene cierta ortogonalidad, pero no completa:

* Algunas construcciones pueden comportarse de forma inesperada (por ejemplo, coerción de tipos con ==).
* La combinación de funciones, objetos, y estructuras de control es flexible, pero no siempre predecible.
* En el paradigma estructurado, si se evita el uso de objetos y funciones de orden superior, la ortogonalidad mejora.

5) ¿Cuáles son sus características de consistencia y uniformidad?

* Moderada consistencia:
  + La sintaxis básica (if, for, function) es uniforme y clara.
  + Sin embargo, hay inconsistencias históricas (como var vs let/const, o el comportamiento de this) que pueden confundir.
  + En el enfoque estructurado, al evitar patrones complejos, se puede mantener una alta uniformidad en el estilo de código.

6) ¿Es extensible? ¿Hay subconjuntos de ese lenguaje?

Sí, JavaScript es altamente extensible:

* Se pueden definir nuevas funciones, módulos, y bibliotecas fácilmente.
* Hay subconjuntos como:
  + Strict Mode ('use strict') que restringe ciertas prácticas peligrosas.
  + TypeScript como superconjunto que añade tipado estático.
  + Vanilla JS como subconjunto que evita frameworks y librerías externas.

7) ¿El código producido es transportable?

Sí, extremadamente transportable:

* JavaScript corre en cualquier navegador moderno, sin necesidad de compilación.
* También puede ejecutarse en servidores (Node.js), dispositivos móviles (React Native), y entornos embebidos.
* Su ubicuidad lo convierte en uno de los lenguajes más portables del mundo.