

## Paradigmas de Programación

### Ejercicio 1

Considera el lenguaje JavaScript acotado al paradigma de programación orientada a objetos basado en prototipos y analízalo en términos de los cuatro componentes de un paradigma mencionados por Kuhn.

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

Sintaxis Central (Reglas Escritas):

- **Tipado Dinámico y Débil:** Las reglas no exigen la declaración de tipos de variables, y el lenguaje realiza conversiones de tipo implícitas.
- **Estructuras de Control:** La sintaxis estándar para if/else, for, while, switch.
- **Definición de Funciones:** Reglas para declarar funciones (function, funciones flecha =>).
- **Sintaxis de Clases:** Aunque se basa en prototipos, la sintaxis de class, constructor, extends, super, static define las reglas para la programación orientada a objetos moderna en JS (a partir de ECMAScript 2015/ES6).
- **Manejo de Prototipos:** La existencia y el uso del objeto prototype y la cadena de prototipos. La sintaxis para la creación de objetos mediante literales ({} ) o funciones constructoras.
- **Asincronía y Promesas:** Reglas sintácticas para el manejo de la asincronía, como Promise, async y await.

La fuente más formal de estas reglas es el estándar **ECMAScript (ES)**, que es la especificación técnica en la que se basa JavaScript. Este documento define exhaustivamente la sintaxis, la semántica y las librerías estándar del lenguaje.

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

- La universalidad: JS es el único lenguaje que **todos los navegadores web entienden** (el *frontend*). Además, gracias a **Node.js**, también puede funcionar como el cerebro de tu aplicación en el servidor (*backend*).
  - Los desarrolladores pueden usar un **solo idioma** para todo el proyecto. Esto significa que es más fácil moverse entre el código del navegador y el código del servidor, se aprende un solo lenguaje potente, y se reutilizan equipos y código.
- Rápido y Multitarea (El modelo asíncrono): JS es de un solo hilo (hace una cosa a la vez), pero es súper eficiente. Cuando le pides que espere algo (como datos de un servidor), no se queda quieto. En su lugar, pone esa tarea en espera y sigue haciendo otras cosas. A esto se le llama programación **asíncrona** o modelo basado en eventos.
  - Para tareas de servidor (como manejar muchos usuarios de una web o un chat), es **muy rápido y escalable**. Puede manejar miles de conexiones al mismo tiempo sin colapsar, usando menos recursos que otros lenguajes que usan muchos hilos paralelos.
- Los prototipos (La flexibilidad de los objetos): A diferencia de otros lenguajes (como Java o C++) donde tienes que definir una "clase" o "molde" estricto antes de crear un objeto, JS te permite **crear objetos al vuelo** y hacer que hereden cosas de otros objetos existentes (*prototipos*). Aunque ahora tiene la palabra class (que es una forma más fácil de escribir lo mismo), su base es muy dinámica.
  - Ofrece mucha **flexibilidad y dinamismo**. Puedes cambiar y adaptar objetos fácilmente mientras el programa se está ejecutando.

- Amplio Ecosistema y Comunidad: JavaScript tiene el ecosistema de herramientas (librerías, *frameworks* como React, Angular, Vue) y la **comunidad de desarrolladores más grande** del mundo.
  - Si tienes un problema, **alguien ya lo resolvió** y lo publicó. Hay una enorme cantidad de herramientas gratuitas, tutoriales y ayuda disponible, lo que acelera mucho el desarrollo.