Paradigmas de programación

Maciel Vigneau

Comision 2

# **Ejercicio 1**

Considera el lenguaje TypeScript acotado al paradigma de programación estructurada y analízalo en términos de [los cuatro componentes de un paradigma](https://www.notion.so/f36d432c55274b93913dc289446f424d?pvs=21) mencionados por Kuhn.

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

RSPT: En el caso de TypeScript, al ser un super set de JavaScript, tiene casi las misma reglas y sintaxis, pero su uso se da para manejar mejor la gestion de logica y datos del programa.

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

RSPT: tiene varios aspectos en los que lo resaltan...

Uno de ellos, es el **tipado estático.**.. que aveces(pongo un ejemplo) al usar objetos como funciones... en JavaScript lo ejecuta bien o mal, lo ejecuta igual. En cambio TypeScript detecta estos errores antes de la ejecución.

**Diseño por Contrato (Interfaces)**: Aunque las interfaces se usan en POO, en el paradigma estructurado se valoran porque definen con precisión los contratos de datos que las funciones deben aceptar o devolver, aumentando la fiabilidad modular.

También, la **Compatibilidad con JS:** Se valora que, a pesar de añadir tipado, TypeScript compila a JavaScript limpio, lo que garantiza que el código final se ejecute en cualquier entorno JS (navegadores, Node.js) y **Herramientas Superiores (IntelliSense):** La información de tipos permite a los IDE ofrecer un **mejor auto-completado** y capacidades de refactorización más seguras que en JavaScript puro, lo que aumenta la productividad.