

Taller
Comparación de Lenguajes
¿Cuál es el mejor?
By Juan Gui

Realizar un comparativo entre los lenguajes, Java, JavaScript, Python y Scala, teniendo en cuenta su rendimiento, adaptación del lenguaje y su seguridad. Realizar esta comparación desarrollando el siguiente ejercicio.

Planificador Funcional de Compras Semanales

Descripción del problema.

Un JuanGui como usuario quiere organizar sus compras semanales según un menú predefinido y un inventario actual. El sistema debe generar automáticamente una lista de compras, eliminando duplicados, agrupando cantidades y priorizando los ingredientes faltantes.

Alcance

Implementar una aplicación que:

1. Reciba un **menú semanal** (lista de recetas por día).
2. Verifique los **ingredientes disponibles** en inventario.
3. Calcule la **lista de compras** con cantidades agregadas de lo que falta.
4. Opcional: clasifique los productos por categoría (e.g., frutas, carnes, lácteos).

Datos de entrada

Menú semanal: Estructura JSON / lista

```
[
  {
    "dia": "lunes",
    "receta": "ensalada",
    "ingredientes": [
      { "nombre": "lechuga", "cantidad": 1, "unidad": "pieza" },
      { "nombre": "tomate", "cantidad": 2, "unidad": "pieza" }
    ]
  },
  {
    "dia": "martes",
    "receta": "pasta",
    "ingredientes": [
      { "nombre": "pasta", "cantidad": 200, "unidad": "gramos" },
      { "nombre": "tomate", "cantidad": 1, "unidad": "pieza" }
    ]
  }
]
```

Inventario disponible:

```
[
  { "nombre": "tomate", "cantidad": 1, "unidad": "pieza" },
  { "nombre": "pasta", "cantidad": 100, "unidad": "gramos" }
]
```

Se espera como Salida: Lista de compras

```
[
  { "nombre": "lechuga", "cantidad": 1, "unidad": "pieza" },
  { "nombre": "tomate", "cantidad": 2, "unidad": "pieza" },
  { "nombre": "pasta", "cantidad": 100, "unidad": "gramos" }
]
```

Nota: se agregaron las cantidades necesarias de lo que no está o no alcanza.

Para la comparación entre los lenguajes podrían usar:

Lenguaje	Consideraciones funcionales
Python	map, flatmap filter, functools.reduce, dataclass, collections.Counter
Scala	case class, map, groupBy, foldLeft, Option, immutable.Map
Java	Stream / Stream API, Collectors.groupingBy, funciones lambda ()
JavaScript	Array.map, flatmap reduce, Set, Object, y opcionalmente Ramda o Lodash FP

Entregar:

1. Implementación en 3 lenguajes.
2. Sustentación del manejo funcional de los desarrollos.
3. Conclusión del por qué el lenguaje escogido fue el ganador.
4. Presentar sustentación frente al rendimiento y seguridad del lenguaje escogido.
5. Opcional: interfaz mínima (CLI o Web).
6. Opcional: Se recomienda usar herramientas como SonarQube conjunto con SonarAnalyzer, Jmeter / Loadero

Nota: JavaScript solo se referencia de forma didáctica, por si se necesita otro lenguaje como comparativa, sin embargo es opcional su desarrollo.