

# Guia Rapida: Estructuras de Datos en PowerShell

PowerShell ofrece diversas estructuras de datos para almacenar, procesar y organizar información.

Esta guía presenta las más comunes y prácticas para el desarrollo de scripts profesionales.

## 1. Comparativa de estructuras de datos

Estructura	Duplicados	Orden	Claves unicas
Array (@(...))	Sí	Sí	N/A
ArrayList	Sí	Sí	N/A
HashSet	No	No garantizado	Sí
Hashtable	Sí (valores)	No garantizado	Sí (claves)
OrderedDictionary	Sí	Sí	Sí
Dictionary<TKey,TValue>	Sí	No garantizado	Sí
PSCustomObject	Sí	N/A	N/A
DataTable	Sí	Sí (filas)	N/A
Queue / Stack	Sí	Sí	N/A
List<T>	Sí	Sí	N/A

## 2. Uso recomendado por escenario

- Array: listas simples y fijas.
- ArrayList: listas modificables, crecimiento dinámico.
- HashSet: lista de valores únicos sin orden.
- Hashtable: clave-valor con claves únicas.
- OrderedDictionary: clave-valor con orden de inserción.
- Dictionary<TKey,TValue>: estructuras robustas y tipadas.
- PSCustomObject: representar objetos con propiedades.

- DataTable: simula una tabla tipo base de datos.
- Queue: estructura FIFO (primero en entrar, primero en salir).
- Stack: estructura LIFO (último en entrar, primero en salir).
- List<T>: listas genéricas más controladas.

### **3. Tips y buenas practicas**

- Usa HashSet si necesitas unicidad automática.
- Usa Hashtable o Dictionary si requieres acceso rápido por clave.
- Usa PSCustomObject para representar entidades y exportar a CSV o JSON.
- Usa Queue/Stack para flujos secuenciales o reversibles.
- Usa DataTable si necesitas procesar datos tabulares o para interoperabilidad.
- Evalúa rendimiento: ArrayList es más rápido que Array cuando se modifica con frecuencia.