Guia Rapida: ArrayList en PowerShell

ArrayList es una estructura de coleccion dinamica de .NET que permite agregar, eliminar, insertar y recorrer elementos de manera flexible y eficiente. Es ideal para trabajar con listas grandes o modificables.

1. Crear un ArrayList

```
$lista = [System.Collections.ArrayList]@()
$lista = [System.Collections.ArrayList]@("manzana", "pera", "uva")
```

2. Agregar elementos - .Add()

```
$lista.Add("mango")  # Agrega al final
```

3. Insertar elementos - .Insert(index, valor)

```
$lista.Insert(1, "naranja")  # Inserta en la posicion 1
```

4. Eliminar elementos - .Remove() y .RemoveAt()

```
$lista.Remove("pera")     # Elimina la primera coincidencia de "pera"
$lista.RemoveAt(2)        # Elimina el elemento en el indice 2
```

5. Eliminar todos los elementos iguales

```
while ($lista.Contains("pera")) {
    $lista.Remove("pera")
}
```

6. Recorrer elementos - foreach y for

```
foreach ($item in $lista) {
```

```
    Write-Host $item
}

for ($i = 0; $i -lt $lista.Count; $i++) {
    Write-Host "$i) $($lista[$i])"
}
```

7. Otros metodos utiles

```
$lista.Clear()          # Elimina todos los elementos
$lista.Contains("uva")   # Retorna True si existe el valor
$lista.IndexOf("mango")  # Devuelve el indice de "mango"
```

8. Tips y buenas practicas

- ArrayList es mas eficiente que += en arrays normales.
- Ideal para listas en crecimiento o modificacion.
- .Remove() elimina solo la primera coincidencia.
- Usa while con .Contains() para eliminar todos los duplicados.
- Se puede combinar con estructuras como foreach y for para iterar.
- Siempre valida el indice antes de usar .RemoveAt() para evitar errores.