

# Guía de lenguaje

## Wollok

### Elementos Comunes

#### Sintaxis básica

Comentario	// un comentario /* un comentario multilínea */
Strings	"uNa CadEna" 'uNa CadEna'
Caracteres	NA
Símbolos/Átomos	NA
Booleanos	true false
Set	#{} #{1, "hola"}
Lista	[] [1, "hola"]
Patrones de listas	NA
Tuplas	NA
Data/Funtores	NA
Bloques sin parámetros	{algo}
Bloques / Exp. lambda (De un parámetro)	{x => algo con x}
Bloques / Exp. lambda (Más de un parámetro)	{x, y => algo con x e y}
Variable anónima	NA

#### Operadores lógicos y matemáticos

Equivalencia	==
Identidad	===
~ Equivalencia	!=
Comparación de orden	> >= < <=
Entre valores	NA
Disyunción (O lógico)	 or
Conjunción (Y lógico)	&& and

Negación	<code>! unBool</code> <code>unBool.negate()</code> <code>not unBool</code>
Operadores aritméticos	<code>+ - * /</code>
División entera	<code>dividendo.div(divisor)</code>
Resto	<code>dividendo % divisor</code>
Valor absoluto	<code>unNro.abs()</code>
Exponenciación	<code>base ** exponente</code>
Raíz cuadrada	<code>unNro.squareRoot()</code>
Máximo entre dos números	<code>unNro.max(otroNro)</code>
Mínimo entre dos números	<code>unNro.min(otroNro)</code>
Par	<code>unNro.even()</code>
Impar	<code>unNro.odd()</code>

## Mensajes de colecciones simples sin efecto

Longitud	<code>coleccion.size()</code>
Si está vacía	<code>coleccion.isEmpty()</code>
Concatenación	<code>coleccion + otraColeccion</code>
Unión	<code>set.union(coleccion)</code>
Intersección	<code>set.intersection( coleccion)</code>
Acceso por índice	<code>lista.get(indice)</code> (base 0)
Pertenencia	<code>coleccion.contains(elem)</code>
Máximo	<code>coleccionOrdenable.max()</code>
Mínimo	<code>coleccionOrdenable.min()</code>
Sumatoria	<code>coleccionNumerica.sum()</code>
Aplanar	<code>coleccionDeColecciones. flatten()</code>
Primeros n elementos	<code>lista.take(n)</code>
Primer elemento	<code>lista.head()</code> <code>lista.first()</code>
Último elemento	<code>lista.last()</code>
Cola	NA
Segmento inicial (sin el último)	NA
Apareo de listas	NA

Sin repetidos	<code>coleccion.asSet()</code>
---------------	--------------------------------

## Mensajes de colecciones con bloques sin efecto

Sumatoria según transformación	<code>coleccion.sum(bloqueNumericoDe1)</code>
Filtrar	<code>coleccion.filter(bloqueBoolDe1)</code>
Transformar	<code>coleccion.map(bloqueDe1)</code>
Todos cumplen (true para lista vacía)	<code>coleccion.all(bloqueBoolDe1)</code>
Alguno cumple (false para lista vacía)	<code>coleccion.any(bloqueBoolDe1)</code>
Transformar y aplanar	<code>coleccion.flatMap(bloqueDe1)</code>
Reducir/plegar a izquierda	<code>coleccion.fold(valorInicial, bloqueDe2)</code>
Reducir/plegar a derecha	NA
Apareo con transformación	NA
Primer elemento que cumple condición	<code>coleccion.find(bloqueBoolDe1)</code> <code>coleccion.findOrElse( bloqueBoolDe1, bloqueSinParametros)</code>
Cantidad de elementos que cumplen condición	<code>coleccion.count(bloqueBoolDe1)</code>
Obtener colección ordenada.	<code>coleccion.sortedBy(bloqueBoolDe2)</code>
Máximo según criterio.	<code>coleccion.max(bloqueOrdenableDe1)</code>
Mínimo según criterio.	<code>coleccion.min(bloqueOrdenableDe1)</code>

## Mensajes de colecciones con efecto

Agregar un elemento.	<code>coleccion.add(objeto)</code>
Agregar todos los elementos de la otra colección	<code>coleccion.addAll(otraColeccion)</code>
Evaluar el bloque para cada elemento.	<code>coleccion.forEach(bloqueConEfectoDe1)</code>
Eliminar un objeto.	<code>coleccion.remove(objeto)</code>
Eliminar elementos según condición	<code>coleccion.removeAllSuchThat(bloqueBoolDe1)</code>
Eliminar todos los elementos.	<code>coleccion.clear()</code>
Deja ordenada la lista según un criterio.	<code>lista.sortBy(bloqueBoolDe2)</code>

## Notas

NA: "No Aplica". No existe o no se recomienda su uso.