Función de String	Funcionamiento
`concat('XML',' y XPath')`	La función `concat()` concatena sus valores de entrada; el caso anterior generaría el String: `'XML y XPath'`.
`contains('osmosislatina','osmosis')`	La función `contains()` es utilizada para verificar si determinado String (primer dato de entrada) contiene cierta secuencia (segundo dato de entrada) ; el ejemplo anterior retornaría verdadero (`true`) ya que `osmosis` se encuentra en el String `osmosislatina`; NOTA: El resultado de esta función siempre da un resultado booleano (`true`/`false`) .
`normalize-space(' Uso de XPath ')`	El método `normalize-space()` es utilizado para eliminar los espacios al inicio y fin de un String; el caso anterior modifica el String a simplemente `'Uso de XPath'`.
`string-length('CentroRecreativo')`	La función `string-length()` genera el numero de caracteres del String indicado; la declaración anterior resulta en 16.
`substring('CentroRecreativo', 1, 6)`	La función `substring()` genera una secuencia a partir del String proporcionado (primer dato de entrada), partiendo de un carácter (segundo dato de entrada) y terminando en otra posición (tercer datos de entrada) ; el ejemplo anterior genera el String `Centro`.
`substring-after('3838-4321','-')`	La función `substring-after` genera una secuencia con los caracteres posteriores a determinado String partiendo de otro String original; el caso anterior generaría el String: `4321`.
`substring-before('5555-1234','-')`	La función `substring-before` genera una secuencia con los caracteres previos a determinado String partiendo de otro String original; el caso anterior generaría el String: `5555`.
`translate('todo','to','na')`	La función `translate()` convierte todos los elementos en determinado String (primer dato de entrada), basado en una secuencia original (segundo dato de entrada) hacia caracteres alternos (tercer dato de entrada) ; el caso anterior generaría el String: `'nada'`, todo carácter `t` es substituido por `n` y todo carácter `o` por `a` .

Función/ Expresión Numérica	Funcionamiento
`+`	Representa el signo para realizar una suma
`-`	Representa el signo para realizar una resta
`*`	Representa el signo para realizar una multiplicación
`div`	El vocablo `div` es utilizado para realizar una división

`12 mod 5`	`mod` representa el residuo de una operación ; la declaración anterior generaría 2.
`ceiling(3.2)`	La función `ceiling()` genera el *techo* ("ceiling") de un numero decimal, en términos matemáticos esto implica el numero entero superior al presente; el ejemplo anterior generaría `4`
`floor(6.7)`	La función `floor()` genera el *piso* ("floor") de un numero decimal, en términos matemáticos esto implica el numero entero inferior al presente; el ejemplo anterior generaría `6`
`number('7')`	La función `number()` convierte un String a número para que puedan ser realizadas operaciones matemáticas.
`round(9.3)`	La función `round` redondea un numero, en términos matemáticos esto implica el numero entero inferior al presente si la decimal es menor a `.5` , o bien, el numero entero superior al presente si su decimal es `.5` o superior; el ejemplo anterior generaría `9`
`sum(Miembro/Casilla)`	La función `sum()` realiza una sumatoria sobre los elementos proporcionados; la declaración anterior generaría la sumatoria de todos los valores presentes en los elementos `Casilla`.

Función/ Expresión Condicional (Booleana)	**Funcionamiento**
`not(`*`expresión`*`)`	La función `not()` genera la negación de la expresión proporcionada, esto es, si la expresión resulta verdadera (`true`) la función generaría un resultado falso (`false`), y viceversa.
`!=`	Representa una desigualdad entre dos elementos, si ambos elementos son iguales resulta verdadera (`true`), caso contrario false (`false`)
`<`	Representa una comparación menor que entre dos elementos, resultando en un valores verdadero (`true`) o falso (`false`) según la comparación.
`<=`	Representa una comparación menor que o igual entre dos elementos, resultando en un valores verdadero (`true`) o falso (`false`) según la comparación.
`>`	Representa una comparación mayor que entre dos elementos, resultando en un valores verdadero (`true`) o falso (`false`) según la comparación.
`>=`	Representa una comparación mayor que o igual entre dos elementos , resultando en un valores verdadero (`true`) o falso (`false`) según la comparación.
`or`	Representa una condicional entre dos elementos, si cualquiera de estos elementos es verdadero (`true`) la condicional también resulta verdadera (`true`), en caso contrario resulta falsa (`false`).
`and`	Representa una unión entre dos elementos, solo sí ambos elementos son verdaderos (`true`) resulta verdadera (`true`) la unión, en caso contrario la unión resulta falsa (`false`).