

ACTIVIDADES A REALIZAR

Actividad 1:


En la carpeta de Rules&Constants se crea una subcarpeta para almacenar expresiones en forma de práctica

Create Rule Folder

Name*

Description

Parent

 AX Rules & Constant ✕ ☰

Dentro de la carpeta de reglas creada, se puede agregar una regla de expresión

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name*

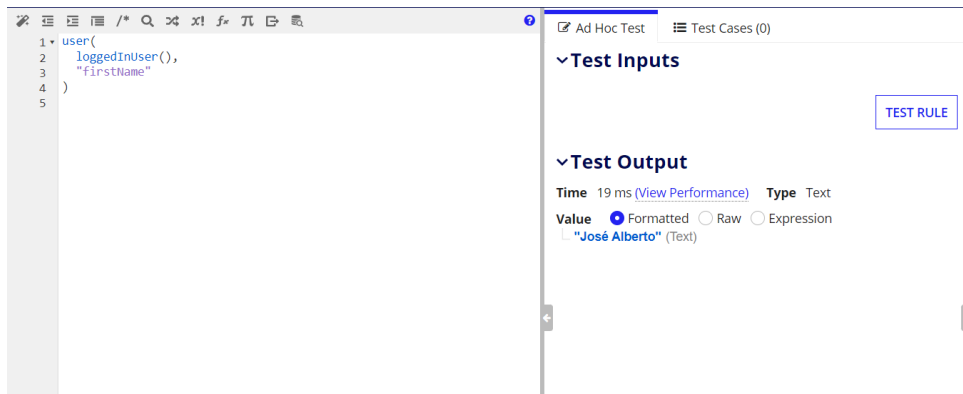
Description

Save In*

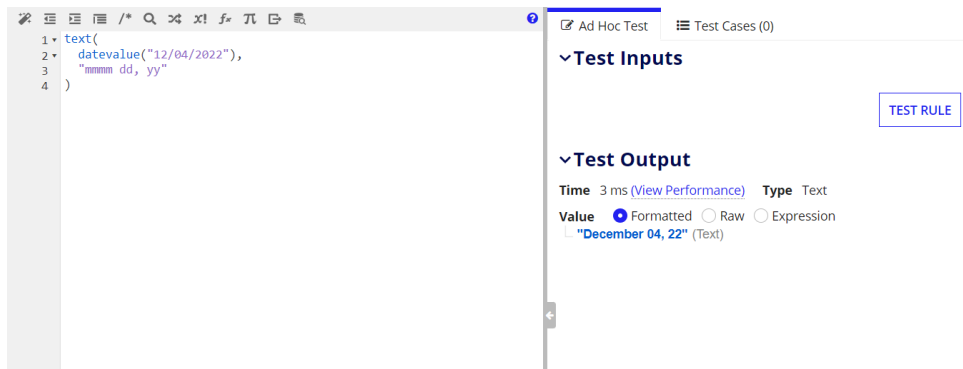
 AX Expressions Practice ✕ ☰

[Create New Rule Folder](#)

El ejercicio consiste en crear un objeto usuario para mostrar el nombre del mismo. Quedando de la siguiente forma



Otro ejercicio para convertir el formato de la fecha



Actividad 2:

Se crea la regla de expresión para poder trabajar sobre ella

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name *

AX_L4Ex1

Description

Practice for Lesson 4

Save In *

AX Expressions Practice x



Create New Rule Folder

CANCEL

CREATE

Probando las condicionales

The screenshot shows a testing tool interface. On the left, a code editor contains the following JavaScript code:

```

1- if(
2-   100 > 200,
3-   "Vendido",
4-   "No vendido"
5- )

```

On the right, the 'Test Results' panel shows:

- Test Inputs:** (Empty)
- Test Output:**
 - Time: < 1 ms (View Performance)
 - Type: Text
 - Value: Formatted (selected), Raw, Expression
 - Value: "No vendido" (Text)

Otro ejercicio consiste en calcular los días enteros entre una fecha y otra

The screenshot shows a testing tool interface. On the left, a code editor contains the following JavaScript code:

```

1- calworkdays(
2-   datetime(2022,01,30),
3-   datetime(2022,05,30)
4- )

```

On the right, the 'Test Results' panel shows:

- Test Inputs:** (Empty)
- Test Output:**
 - Time: 56 ms (View Performance)
 - Type: Number (Integer)
 - Value: Formatted (selected), Raw, Expression
 - Value: 86 (Number (Integer))

Por último, un ejercicio para verificar que una cadena de texto no sobrepase la cantidad límite de caracteres que se establece

The screenshot shows a testing tool interface. On the left, a code editor contains the following JavaScript code:

```

1- if(
2-   len(!texto_prueba) > maxvalor,
3-   "No debe pasar los " & !maxvalor & " caracteres",
4-   "Gracias"
5- )

```

Below the code editor, there is a documentation snippet:

```

!(condition, valueIfTrue, valueIfFalse)
Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.
Returns: Any Type
condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.
valueIfTrue (Any Type): The value to be returned if condition returns true.
valueIfFalse (Any Type): The value to be returned if condition returns false.

```

On the right, the 'Test Case' configuration panel shows:

- Test Inputs:**

Rule Input Name	Expression	Value
texto_prueba (Text)	1	HolaMundo
maxvalor (Number (Integer))	1	8
- Test Output:**
 - Time: 1 ms (View Performance)
 - Type: Text
 - Value: Formatted (selected), Raw, Expression
 - Value: "No debe pasar los 8 caracteres" (Text)

At the bottom right, there is a button that says "Open the Learning Center to select a tutorial or video" with a question mark icon.

Actividad 3:

Se crea la regla de expresión para poder trabajar sobre ella

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name *

AX_L5Ex1

Description

Practice for Lesson 5

Save In *

AX Expressions Practice

Create New Rule Folder

CANCEL

CREATE

La primera actividad consiste en probar las condicionales usando los Rules Input, si dos números se exceden por cierto valor, mostrar un mensaje u otro

AX_L5Ex1

```
1 - if(  
2 * (if(fin-ri[inicio])>50000,  
3 "Retornar diagnostico completo",  
4 "Retornar inventario"  
5 )
```

if(condition, valueIfTrue, valueIfFalse)
Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.
Returns: Any Type
condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.
valueIfTrue (Any Type): The value to be returned if condition returns true.
valueIfFalse (Any Type): The value to be returned if condition returns false.

Rule Input Name	Expression	Value
inicio (Number (Integer))	1	10000
fin (Number (Integer))	1	50000

Save as Test Case TEST RULE

Time 1 ms (View Performance) Type Text
Value Formatted Raw Expression
"Retornar inventario" (Text)

Open the Learning Center to select a tutorial or video

Otro ejercicio es encontrar la suma de dos valores mediante variables locales

AX_L5Ex2

```
1 - allLocalVariables(  
2 localSum:ri[primer+ri[segundo,  
3 "La suma es " & localSum  
4 )
```

allLocalVariables localVar1, localVarN, expression
Lets you define one or more local variables for use within an expression. When used within an interface, the value of each variable can be refreshed under a variety of conditions. configured using alrefreshVariable(). When used outside of an interface, all refresh properties configured using alrefreshVariable() are ignored.
localVar1 (Any Type): The local variable to use when evaluating the given expression. Use the 'local:' domain to define and reference individual variables. By default, a local variable will automatically update when any variables it references are changed. To change the way variables are updated, use the alrefreshVariable() function. Variables can be refreshed under the following conditions: after each reevaluation, periodically on an interval, or when other variables change.
localVarN (Any Type): An unlimited number of local variables.

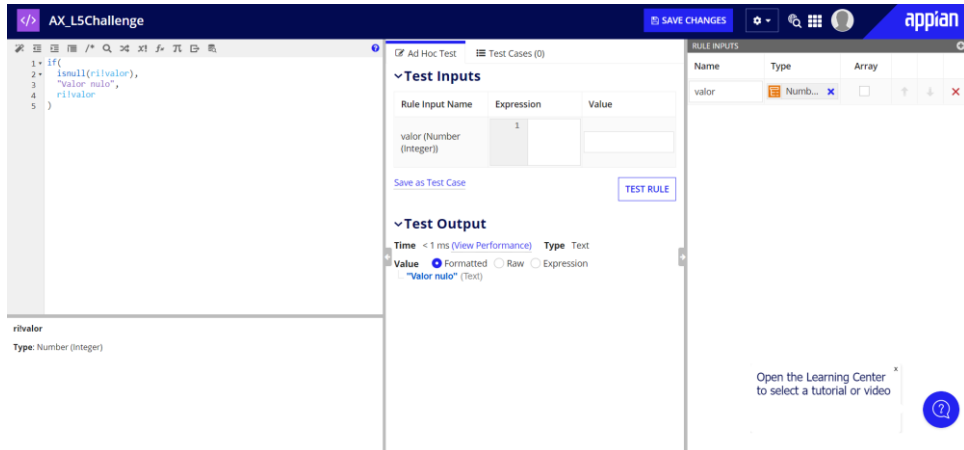
Rule Input Name	Expression	Value
primero (Number (Integer))	1	10
segundo (Number (Integer))	1	30

Save as Test Case TEST RULE

Time 1 ms (View Performance) Type Text
Value Formatted Raw Expression
"La suma es 40" (Text)

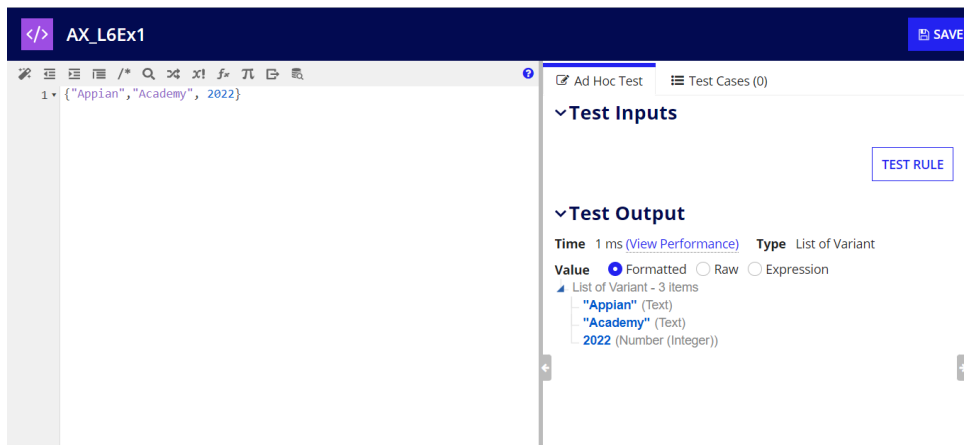
Open the Learning Center to select a tutorial or video

Por último ejercicio, identificar si una variable es nula o no

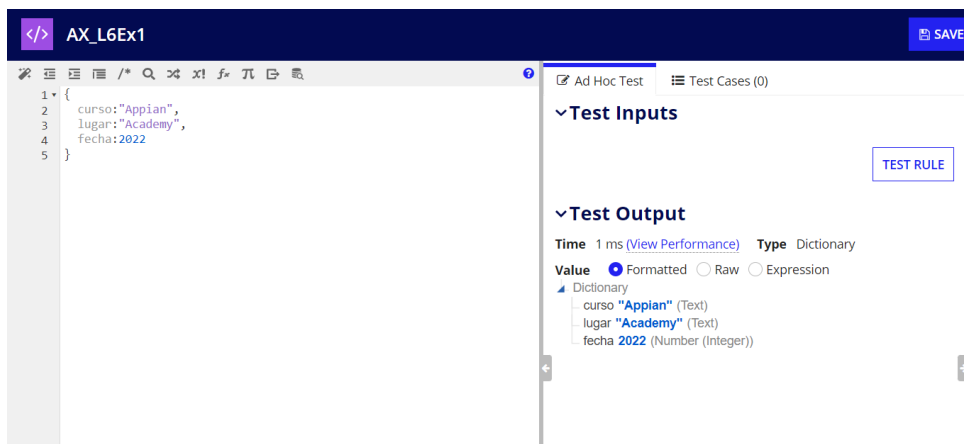


Actividad 4:

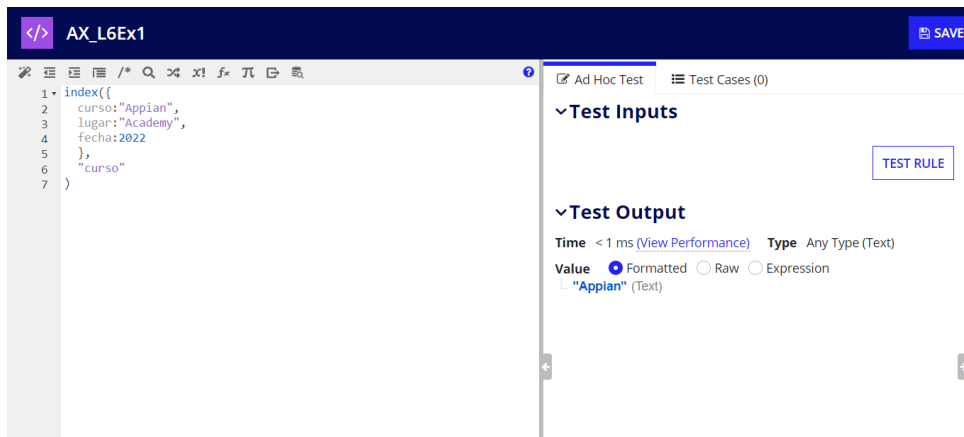
Para esta actividad se crea un arreglo de objetos y se prueba



El arreglo se convierte en diccionario cuando se agrega la llave a la que se relaciona



Y por buenas prácticas se usa el index()

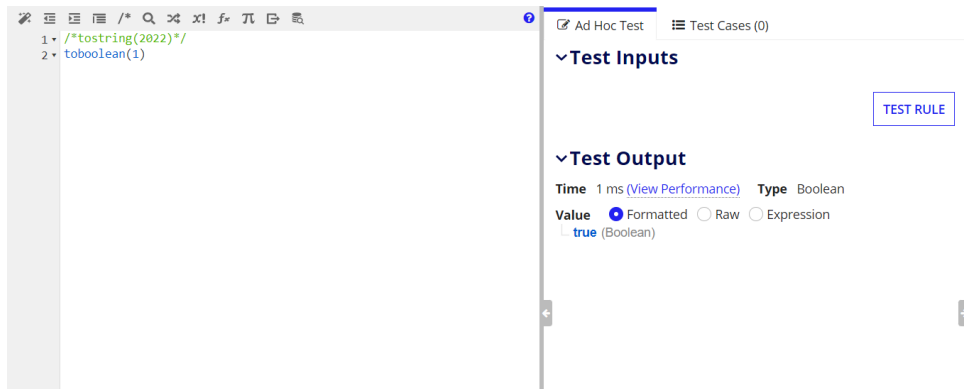


Actividad 5:

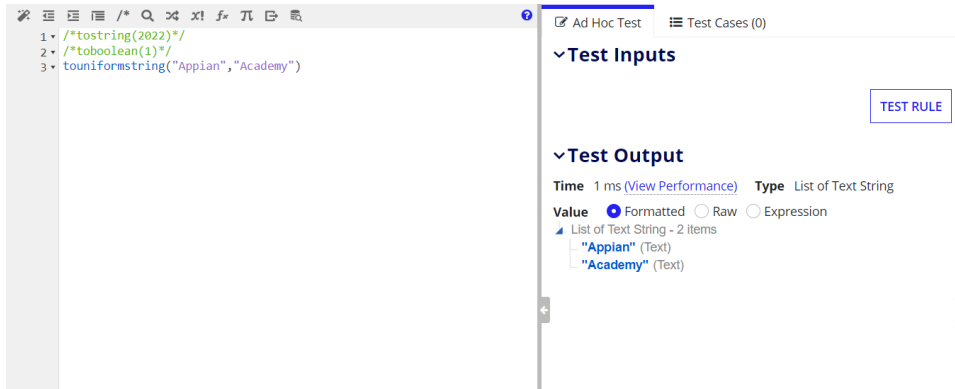
Convertir un número a texto



Convertir un número a booleano

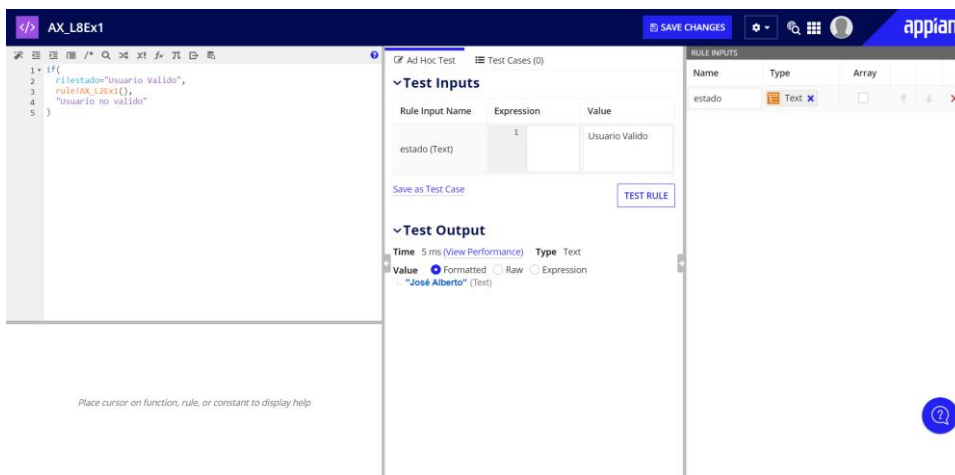


Convertir texto separado por comas en una lista de textos

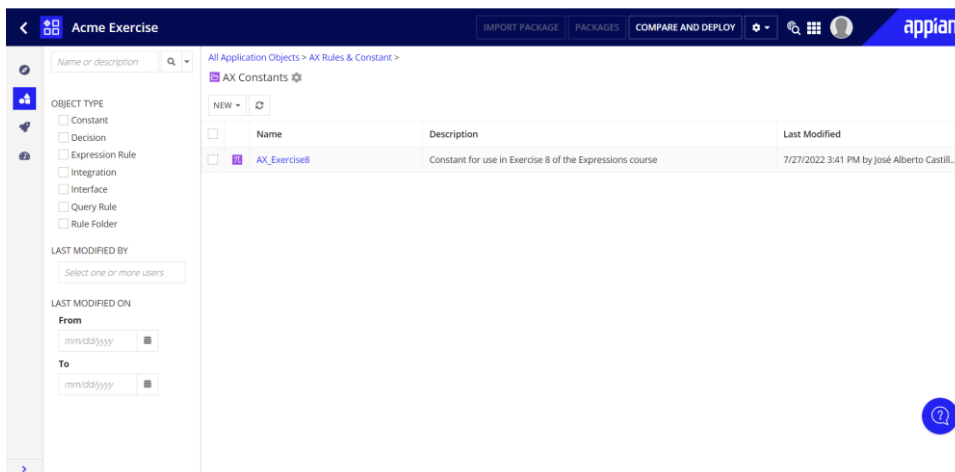


Actividad 6:

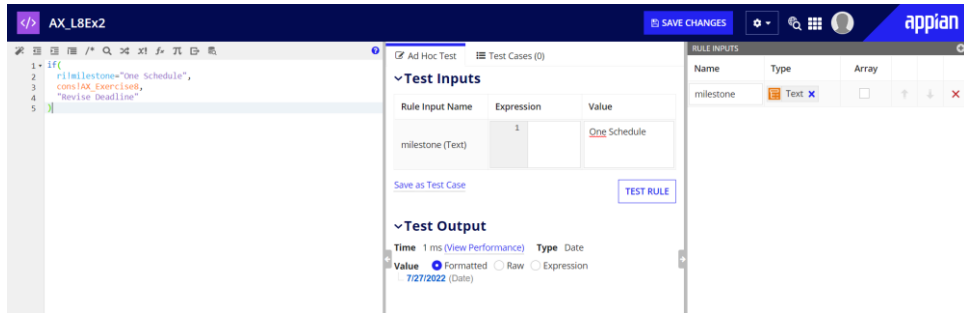
Hacer llamado a la regla de expresión del ejercicio 2



Ahora se creará una constante, que almacenará una fecha y se guardará en AX Constants

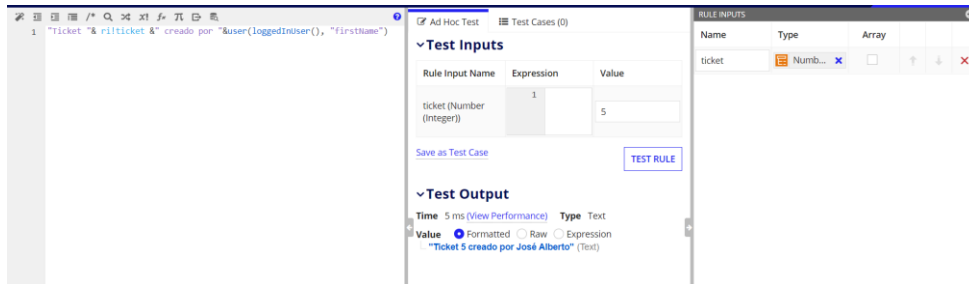


Hacer llamado a la constante creada.



Actividad 7:

Mostrar mensaje de ticket



RESUMEN DE LAS ACTIVIDADES

