

# Laboratorio 6

## Contenidos

|                                       |   |
|---------------------------------------|---|
| Objetivos.....                        | 2 |
| Creación Process Model .....          | 2 |
| Configuración del Process Model ..... | 2 |
| Añadir input task y configurar .....  | 4 |
| Conclusiones .....                    | 9 |

## Objetivos

En este laboratorio crearemos un simple modelo de proceso, en el que se incluirá una llamada al interfaz creado en el paso anterior (XXX\_ReviewVehicleForm).

Este proceso nos servirá para ver como conectar distintos procesos.

**Nota:** Si no pudimos completar el laboratorio anterior, podemos hacer uso del interfaz ya existente en la aplicación con la solución (Open\_Vehicle Fleet Management). En este caso, el interfaz a emplear sería Open\_ReviewVehicleForm.

## Creación Process Model

### Descripción:

Crearemos nuestro Process Model simple, en el que únicamente incluiremos una tarea humana para llamar al interfaz de aprobación por parte del supervisor.

### Pasos a realizar:

- Dentro de nuestra aplicación XXX\_Vehicle Fleet Management, en la carpeta Process creada con anterioridad, pulsamos New -> Process Model.
  - **Name:** XXX\_Review\_Vehicle
  - **Description:** Process Model para para tratar la aprobación de vehículos por el supervisor.
  - **Save in:** XXX\_Process
  - Create.

## Configuración del Process Model

### Descripción:

Una vez creado nuestro proceso XXX\_Review\_Vehicle, pulsamos sobre él para que se nos abra el entorno Appian Process Modeler para implementarlo. En primer lugar, procederemos a configurar su información propia

### Pasos a realizar:

- En el menú superior, pulsamos sobre el icono de Properties.
- Se nos abrirá una ventana emergente, con varias pestañas.
- Pestaña Variables: son las variables que necesitaremos manejar dentro de nuestro proceso. Pulsando sobre la opción "Add Variable" añadiremos las siguientes variables:
  - supervisorComment, donde guardaremos los comentarios del supervisor
  - approvalDecision, donde recogeremos la decisión de si se acepta o no el vehículo.

- requestor, variable que usaremos para guardar la persona a la que está asignada la tarea.
- vehicle, donde guardaremos el vehículo.
- Las variables vehicle y requestor las definiremos como parámetro de entrada de nuestro proceso. También debemos definirlos como obligatorios.
- Al finalizar el alta, se nos deben quedar nuestras variables como se muestra en la siguiente imagen:

General

Variables

Process Start Form

Deadlines

Alerts

Data Management

Process Variables

+

Add Variable

| Name              | Type         | Value      | Parameter? | Required? | Multiple? | Hidden? |   |
|-------------------|--------------|------------|------------|-----------|-----------|---------|---|
| approvalDecision  | Text         | (No Value) | No         | N/A       | No        | No      | ✗ |
| requestor         | User         | (No Value) | Yes        | Yes       | No        | No      | ✗ |
| supervisorComment | Text         | (No Value) | No         | N/A       | No        | No      | ✗ |
| vehicle           | Open_vehicle | (No Value) | Yes        | Yes       | No        | No      | ✗ |

Definimos como parámetro de entrada

Definimos como parámetro de entrada

**Nota:** en nuestro caso, la variable vehicle será de tipo XXX\_vehicle

- Pestaña General: en esta pestaña se recogen datos generales del proceso. Únicamente modificaremos un par de campos de esta pestaña:
  - **Process Model Name:** XXX\_Review\_Vehicle
  - **Process Display Name:** ="Review Vehicle LicensePlate "&pv!vehicle.licensePlate
  - **Nota:** no se trata de una errata, el signo igual también debe ser incluido en el campo Process Display Name.
  - El campo Process Display Name es el que usaremos posteriormente en las pruebas para localizar nuestro proceso en la monitorización de las pruebas. Por esto, mostramos la matrícula del vehículo introducido, para así tener una fácil identificación de cada proceso lanzado.
- Pestaña Alertas



El resultado de la pestaña debe ser el siguiente:

**Process Model Properties » Open\_AddVehicle** ✕

General Variables Process Start Form Deadlines **Alerts** Data Management

**Alert Recipients** ⓘ Who will be notified of errors in this process?

☐ Use system defaults for error alerts.  
☒ Use custom error alert settings.

☒ Send alerts to the process initiator.  
☐ Send alerts to the process model owner.  
☐ Send alerts to the recipients defined by this expression:    
☐ Send alerts to the following users & groups:  

Con esto conseguimos que, en el caso de que se produzca un error durante la ejecución del proceso, se enviará un correo de alerta al usuario que lanzó la ejecución

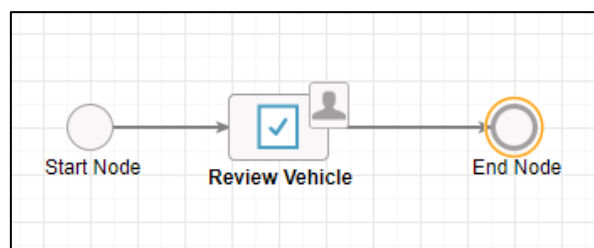
## Añadir input task y configurar

### Descripción:


Añadiremos en nuestro flujo, como primer paso, el interfaz que hemos definido en el laboratorio anterior (XXX\_ReviewVehicleForm).

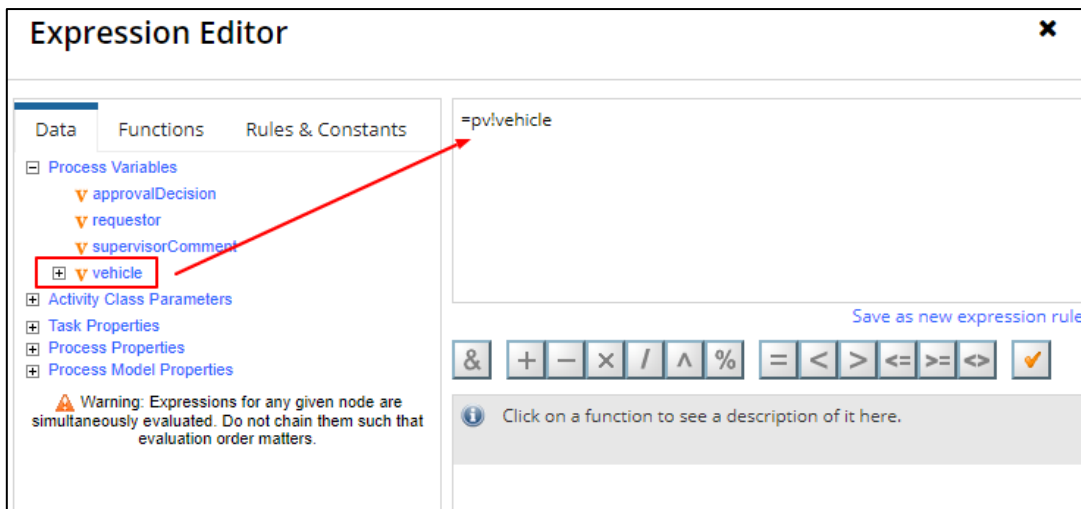
### Pasos a realizar:

- En el menú de la izquierda (Palette), localizar el nodo denominado User Input Task (se encuentra dentro en Standard Nodes - Activities. También existe sobre el árbol un campo a modo de buscador) y arrastrarlo al panel central (canvas), justo sobre la línea que une los eventos Start y End. El resultado debe ser el siguiente:



- Si pulsamos con el botón derecho sobre el icono recién añadido - Properties, se nos muestra una ventana emergente con varias pestañas, con todo lo que podemos gestionar sobre dicho componente.
- Pestaña General
  - Name:** Review Vehicle
  - Task Display Name:** ="Review Vehicle LP"
  - Task Name:** &pv!vehicle.licensePlate

- El campo Task Display Name es el que usaremos posteriormente para localizar nuestro proceso entre las tareas que tengamos asignadas.
- Pestaña Forms: en esta pestaña indicaremos qué interfaz será el que se mostrará en este punto. En nuestro caso, XXX\_ReviewVehicleForm.
- Debemos indicar que sí queremos que nos genere automáticamente los inputs de entrada necesarios.
- **Notas:** nuestra entrada vehicle será de tipo XXX\_Vehicle.
- El valor de la rule input de vehicle lo obtendremos de la variable de proceso (pv) llamada también vehicle, la cual hemos dado de alta en paso 1 de este laboratorio. Para ello, pulsamos sobre el icono  situado a la izquierda de la rule input de vehicle. Se nos abrirá una ventana como la siguiente:



A continuación, una imagen con el resultado:

General Data **Forms** Scheduling Assignment Escalations Exceptions Other

Edit Form

English (US)

☒ Enable this language

☒ Select an interface ☐ Write an expression

"Open\_ApproveVehicle" Clear

**EDIT INTERFACE**

Specify the data to pass to the interface Refresh

| Rule Input        | Type         | Multiple? | Value             |
|-------------------|--------------|-----------|-------------------|
| vehicle           | Open_vehicle | No        | =pvlvehicle       |
| approvalDecision  | Text         | No        | approvalDecision  |
| supervisorComment | Text         | No        | supervisorComment |

☒ Allow users to save a draft of in-progress tasks

- Pestaña Data: Al acceder a la pestaña, observaremos que se han creado 3 inputs de entrada. Debemos modificarles los valores de los campos para que las propiedades de cada una de ellas queden como sigue:

Node Inputs Map the value(s) for the inputs of the node

New Input Delete Input

- vehicle** (Open\_vehicle)
- approvalDecision (Text)
- supervisorComment (Text)

**Field Properties**

Name: vehicle

Type: Open\_vehicle

Multiple: ☐

Value:

Required: ☐

Save into:

**Nota:** nuestro input será de tipo xxx\_vehicle.

Con este input definimos una entrada de tipo vehículo, pero no le asignamos ningún valor.

The screenshot shows the 'Node Inputs' panel on the left and the 'Field Properties' panel on the right. In the 'Node Inputs' panel, the 'approvalDecision' input is selected. The 'Field Properties' panel shows the following configuration:

| Field Properties |                          |
|------------------|--------------------------|
| Name             | approvalDecision         |
| Type             | Text                     |
| Multiple         | <input type="checkbox"/> |
| Value            | =pv!approvalDecision     |
| Required         | <input type="checkbox"/> |
| Save into        | approvalDecision         |

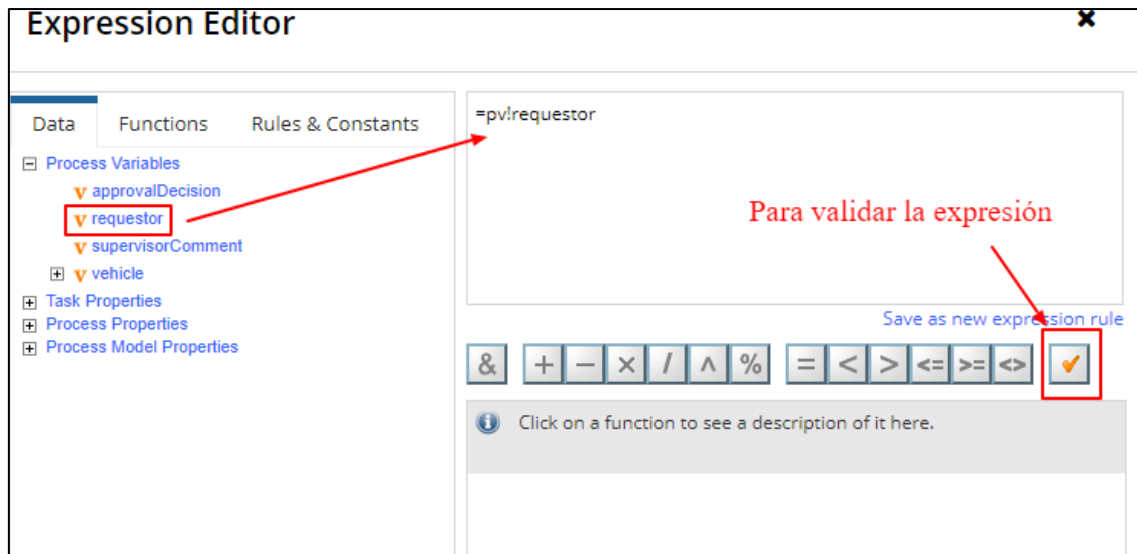
Con este input definimos donde guardar la decisión del aprobador. Tomaremos el valor de la variable de proceso (pv).

The screenshot shows the 'Node Inputs' panel on the left and the 'Field Properties' panel on the right. In the 'Node Inputs' panel, the 'supervisorComment' input is selected. The 'Field Properties' panel shows the following configuration:

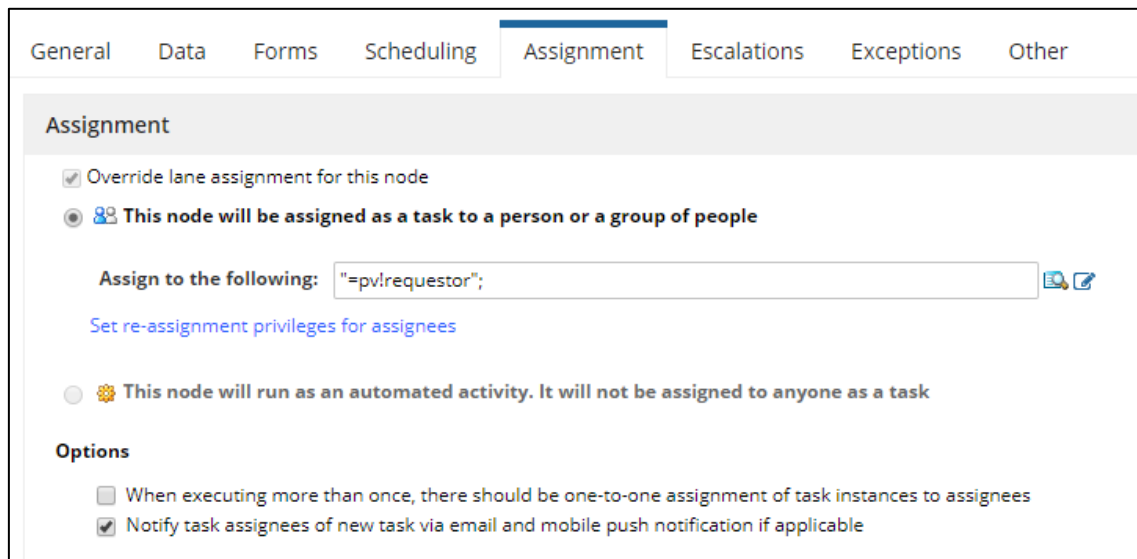
| Field Properties |                          |
|------------------|--------------------------|
| Name             | supervisorComment        |
| Type             | Text                     |
| Multiple         | <input type="checkbox"/> |
| Value            | =pv!supervisorComment    |
| Required         | <input type="checkbox"/> |
| Save into        | supervisorComment        |

Con este input definimos donde guardar los comentarios del aprobador. Tomaremos el valor de la variable de proceso (pv).

- Pestaña Assignment: se indica a quién se asignará la tarea. En nuestro caso, se la asignaremos a la misma persona que inició el proceso. Para ello, dentro de la pestaña Assignment, en el campo Assign to the following, pulsando en el icono "From Expression" que se muestra a la derecha del campo, incluir lo siguiente:



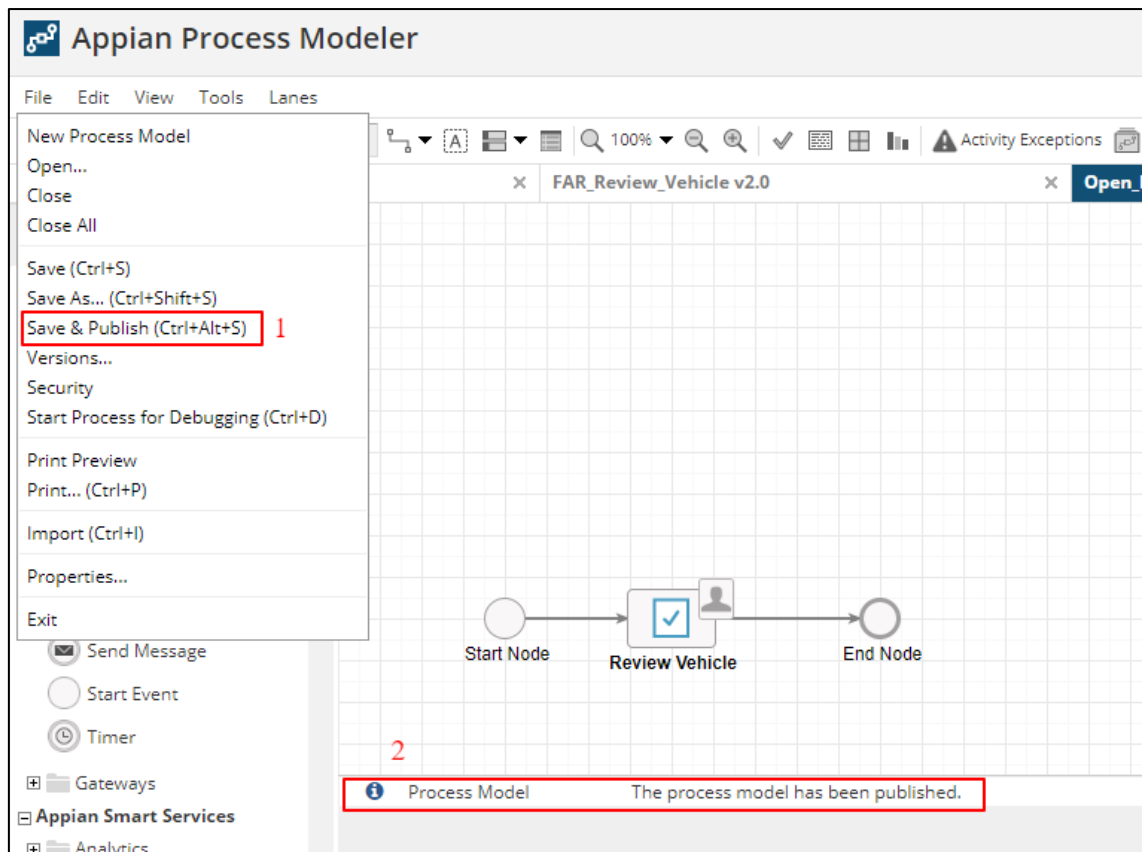
El resultado debería ser el siguiente:



- Pulsamos Save And Close.
- Nos vamos al menú superior – File – Save and Publish.
- Una vez terminada la publicación, nos debe aparecer en la parte inferior del panel central un mensaje informativo indicándonos que el process model se ha publicado con éxito.

Cada vez que realicemos alguna modificación sobre un proceso, debemos volver a publicarlo para que sus cambios sean visibles en las próximas pruebas.





## Conclusiones

En este laboratorio hemos creado un simple proceso que incluye una tarea humana. Este proceso será incluido dentro de otro proceso.

Copyright © 2019 Accenture  
All rights reserved.

Accenture, its Signature, and  
High Performance Delivered  
are trademarks of Accenture.