

# Laboratorio 1

## Contenidos

Objetivos de los laboratorios.....	2
Preámbulos .....	2
Crear Aplicación .....	2
Crear Carpetas.....	3
CDT .....	3
Crear DataStore .....	4
Crear Constante.....	5
Conclusión.....	5

## Objetivos de los laboratorios

Crearemos una pequeña aplicación para la inclusión de vehículos en nuestra flota. Para ello:

- Crearemos nuestra propia aplicación, con una estructura de carpetas para contener los objetos.
- Crearemos una tabla en base de datos (DataStore) donde guardaremos los vehículos.
- Construiremos varias pantallas (interfaces) desde donde los empleados añadirán los vehículos. Le incluiremos varias validaciones para comprobar que los datos contenidos son correctos.
- Construiremos procesos de modelo (Process Model), en el que incluiremos lógica, para decidir, en función de los campos introducidos en los formularios, tomar un camino u otro.

## Preámbulos

- Para la realización de los laboratorios, deberemos crear una serie de objetos. Los nombres de estos objetos han de ser únicos en todo el entorno, por lo que cuando los vayamos a crear, incluiremos como prefijo nuestras propias iniciales. Es decir, si se nos solicita crear una regla de expresión y llamarla XXX\_CalculateNextService, un participante llamado **Luis Garrido Soto** lo llamará **LGS\_CalculateNextService**
- En algunos de los laboratorios, en lugar de partir de un objeto desde 0 (scratch), se parte de un objeto ya creado que se usará como plantilla, con una estructura base, a partir de la cual comenzaremos nuestro desarrollo.
- Las soluciones a todos los ejercicios están ofrecidas dentro de la aplicación **Open\_Vehicle Fleet Management**. Accediendo a dicha aplicación, podrá ver cómo están implementados todos y cada uno de los laboratorios.
- Si en algún momento del laboratorio, no se consigue finalizar alguno de los ejercicios, y este es un paso necesario para la realización del siguiente, podremos crear el objeto que necesitamos en nuestra aplicación a partir del mismo que se encuentra en la aplicación **Open\_Vehicle Fleet Management**. El nombre del objeto que necesitamos copiar tendrá el mismo nombre que el que se nos indique en el laboratorio, pero tendrá como prefijo *Open\_*. Por ejemplo, si se nos pide crear un interfaz llamado XXX\_AddVehicleForm, existirá uno llamado Open\_AddVehicleForm dentro de la aplicación Open\_Vehicle Fleet Management

## Crear Aplicación

### Descripción:

Crearemos nuestra propia aplicación, donde guardaremos todos los objetos que construyamos.

#### Pasos a realizar:

- Navegamos hasta el entorno Design con la URL <https://accenture-openathon.appiancloud.com/suite/design> y pulsamos sobre 'New Application'.
  - **Name:** XXX\_Vehicle Fleet Management.
- Incluir la descripción "Entorno de XXX para la gestión de la flota de vehículos"

## Crear Carpetas

#### Descripción:

Los desarrollos los tendremos organizados por carpetas para tenerlos estructurados. Crearemos 3 carpetas, una para las rules & constantes, para interfaces y para los procesos.

#### Pasos a realizar:

- Dentro de nuestra aplicación recién creada, pulsamos New -> Folder
  - **Type:** Rule Folder
  - **Name:** XXX\_Rules\_And\_Constants
  - **Description:** carpeta para recoger las reglas y constantes
- Dentro de nuestra aplicación recién creada, pulsamos New -> Folder
  - **Type:** Rule Folder
  - **Name:** XXX\_Interfaces
  - **Description:** carpeta para recoger los interfaces
- Dentro de nuestra aplicación recién creada, pulsamos New -> Folder
  - **Type:** Process Model Folder
  - **Name:** XXX\_Process
  - **Description:** carpeta para recoger los modelos de procesos

## CDT

#### Descripción:

Appian puede trabajar con los datos de varias formas, pero una forma muy práctica es agrupar los datos relacionados. Así es más fácil referenciarlos, consultarlos, etc.

Estas agrupaciones lógicas de los datos se llaman Custom Data Type (CDT) o tipo de datos personalizado.

Los CDT se pueden usar para leer y escribir en una tabla de base de datos, para almacenar información dentro de un proceso, o para definir entradas o salidas de un servicio web o plugin de Appian.

Vamos a crear nuestro CDT para contener los datos de un vehículo.

(Nota: para agilizar el laboratorio, vamos a partir de una CDT ya creada)

## Pasos a realizar:

- New -> Data Type
- Duplicate existing data type
  - **Data Type to Duplicate:** CDTTemplate\_Vehicle
  - **Name:** XXX\_Vehicle
- Añadir los siguientes campos a los ya existentes:

Name	Type
vehicleCondition	Text
lastUpdated	Date And Time
nextServiceDate	Date
mileage	Number (integer)
pictureId	Number (integer)

- Pulsamos el botón SAVE.

## Crear DataStore

### Descripción:

Necesitamos un datastore donde crearemos las tablas (entities). Estas tablas serán creadas a partir de nuestros objetos CDT.

## Pasos a realizar:

- New -> DataStore
  - **Name:** XXX\_VFM
  - **Description:** DataStore para la flota de vehículos
- Add Entity
  - **Name:** xxxvehicle
  - **Type:** XXX\_Vehicle
- Pulsamos SAVE
- Pulsamos botón Verify (Está al final de la página). Esta acción sirve crear la tabla (caso de no existir), o para actualizarla, caso de que se haya modificado la CDT asociada. Nos indicará que la tabla no existe. Seleccionamos que se cree automáticamente.
- Save & Publish.

Para verificar la creación correcta de nuestra tabla, vamos al entorno de la base de datos:

<https://accenture-openathon.appiancloud.com/database>

Encontraremos nuestra tabla en el árbol de la izquierda.

En el panel central, podemos movernos por las pestañas y revisar las columnas de nuestra tabla.

## Crear Constante

### Descripción:

Usaremos constantes, para tener valores estáticos guardados y que pueda ser reutilizados donde se requieran.

### Pasos a realizar:

Nota: En este caso, nos apoyaremos en una constante ya existente en el entorno.

- Dentro de nuestra carpeta XXX\_Rules\_And\_Constants, New -> Constant.
- Seleccionamos Duplicate existing constant
  - **Constant to duplicate:** Open\_VEHICLE\_CATEGORIES
  - **Name:** XXX\_VEHICLE\_CATEGORIES
  - **Save In:** XXX\_Rules\_And\_Constants
- El resto de los campos no los modificamos

## Conclusión

En este laboratorio, hemos creado lo necesario para tener una estructura que nos haga posible la elaboración del resto de laboratorios: nuestra propia aplicación, las carpetas para organizar los objetos que creemos, nuestra propia CDT, y su entity en base de datos.

Copyright © 2019 Accenture

All rights reserved.

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