

Trabajo Práctico Integrador

La empresa TuriCor S.A., dedicada a la venta de servicios turísticos en la ciudad de Córdoba, ofrece entre su paquete de servicios el alquiler de vehículos en diferentes destinos turísticos. Actualmente las reservas de cada vehículo se realiza contactando a alguno de sus proveedores únicamente por vía telefónica o vía e-mail. Con cada venta el vendedor debe cargar la información de la reserva en el sistema, generando demoras en cada operación e inconsistencias en algunos casos.

Actualmente, los proveedores más importantes tienen disponible un servicio web que permite la consulta y reserva de vehículos a través de internet.

Con el objetivo de implementar el alquiler de vehículos online en su sistema web de ventas, se requiere implementar la integración a un servicio web (SOAP) de uno de los proveedores para el alquiler de vehículos.

A continuación los datos del servicio web SOAP del proveedor (WCFReservaVehiculos):

URL WSDL

(A definir)

Catálogo de Errores

Método	ID	Descripción
ConsultarVehiculosDisponibles	1	La ciudad no se encuentra registrada.
ReservarVehiculo	21	El vehículo seleccionado no existe.
ReservarVehiculo	22	El vehículo no se encuentra disponible.
CancelarReserva	41	La reserva que se desea cancelar no existe.
CancelarReserva	42	La reserva se encuentra cancelada.
ConsultarCiudades	51	No se encontraron ciudades.
ConsultarPaises	61	No se encontraron países.
Aplica a todos	99	Error no controlado.

Se solicita

1. Desarrollar un servicio rest (Web Api) que cumpla con los siguientes requerimientos:
 - a. Para **Autenticación** de consumo de las APIs se deberá utilizar el estándar OAuth 2.0 , usando el flujo Authorization-Code. La **Autorización** se debe validar usando el campo scope del access-token (con lo cual solo los access token que tengan un determinado scope podrán consumir una API).

Endpoint Authorize:

(A definir)

Endpoint Access-Token:

(A definir)

Endpoint Token-Info:

(A definir)

Datos del usuario final:

usuario=turicor

password=turicor

- b. Consultar los Vehículos disponibles para una ciudad destino consumiendo el servicio SOAP del proveedor. Se debe calcular el precio de venta al público, con una ganancia del 20%.
 - c. Reservar un Vehículo seleccionado (en el proveedor a través del servicio SOAP) y registrar en el sistema de ventas la reserva con los siguientes datos:
 - CódigoReserva: Código de reserva retornado por el servicio.
 - FechaReserva: Fecha de la Reserva.
 - IDCliente: Identificador del cliente que solicitó la reserva.
 - Costo: Corresponde al precio de venta que retorna el servicio WCFReservaVehiculos.
 - PrecioVenta: Precio que incluye el 20 % de ganancia sobre el precio que retorna el servicio WCFReservaVehiculos.
 - d. Consultar listado de reservas realizadas.
 - e. Cancelar una reserva activa.
2. Desarrollar la integración del servicio Rest del punto anterior en un sitio web que tenga las siguientes interfaces:
 - a. Listado de Vehículos disponibles.
 - b. Listado de Reservas realizadas.
 - c. Visualización Reserva.
 - d. Cancelación de Reserva.
3. Documentar el servicio API Rest utilizando Swagger.

Aclaraciones

- Los vehículos se encuentran catalogados por ciudad, por lo tanto la reserva se realiza utilizando un identificador de vehículo por ciudad (IDVehiculoCiudad).
- El servicio rest y la página web deben ser proyectos separados, para evitar acoplamiento.
- Cada grupo se deberá autenticar al servicio web de reserva con sus credenciales, que se les enviará a cada grupo por separado.
- Si bien la persistencia de la reserva se realiza en el servicio WCFReservaVehiculos, existen datos que son necesarios almacenar en una base de datos propia, por lo tanto deben generar base para almacenar los campos que se indican en el punto 1.b. Además, debe existir una tabla de clientes para consultarlos al momento de la reserva.

Presentación

- El trabajo práctico podrá presentarse en grupo de no más de 3 personas.
- El lenguaje a utilizar es a elección de cada grupo.
- Fechas
 - **Primera Fecha: Martes 26/06/18**
 - **Segunda Fecha: Viernes 29/06/18**