

PROYECTO 2016- 2017: DISEÑO INICIAL

Fecha: Marzo de 2017

GRUPO: C3-2

Miembros:

Pérez Vaz, Rubén

Sánchez Piñeiro, Sara

Blanco Caamaño, Marta

Berezo Loza, Sara

Presentación del proyecto

Nuestro servicio web ofrecerá al cliente una plataforma para viajar en avión a los diferentes destinos nacionales (España). Cuando un cliente se conecte, introducirá la ciudad de origen y de destino de su vuelo. Nuestro sistema detectará si existe o no algún aeropuerto en las localidades seleccionadas. En caso afirmativo, devolverá al cliente una lista de vuelos disponibles.

A continuación el cliente tendrá la opción de comprar un billete para uno de esos vuelos, o simplemente salir y utilizar el servicio a modo de consulta. Por último, si escoge la opción de compra, nuestro servicio le enviará una notificación al cliente cuando se haya hecho efectivo el pago.

Diagrama de flujo

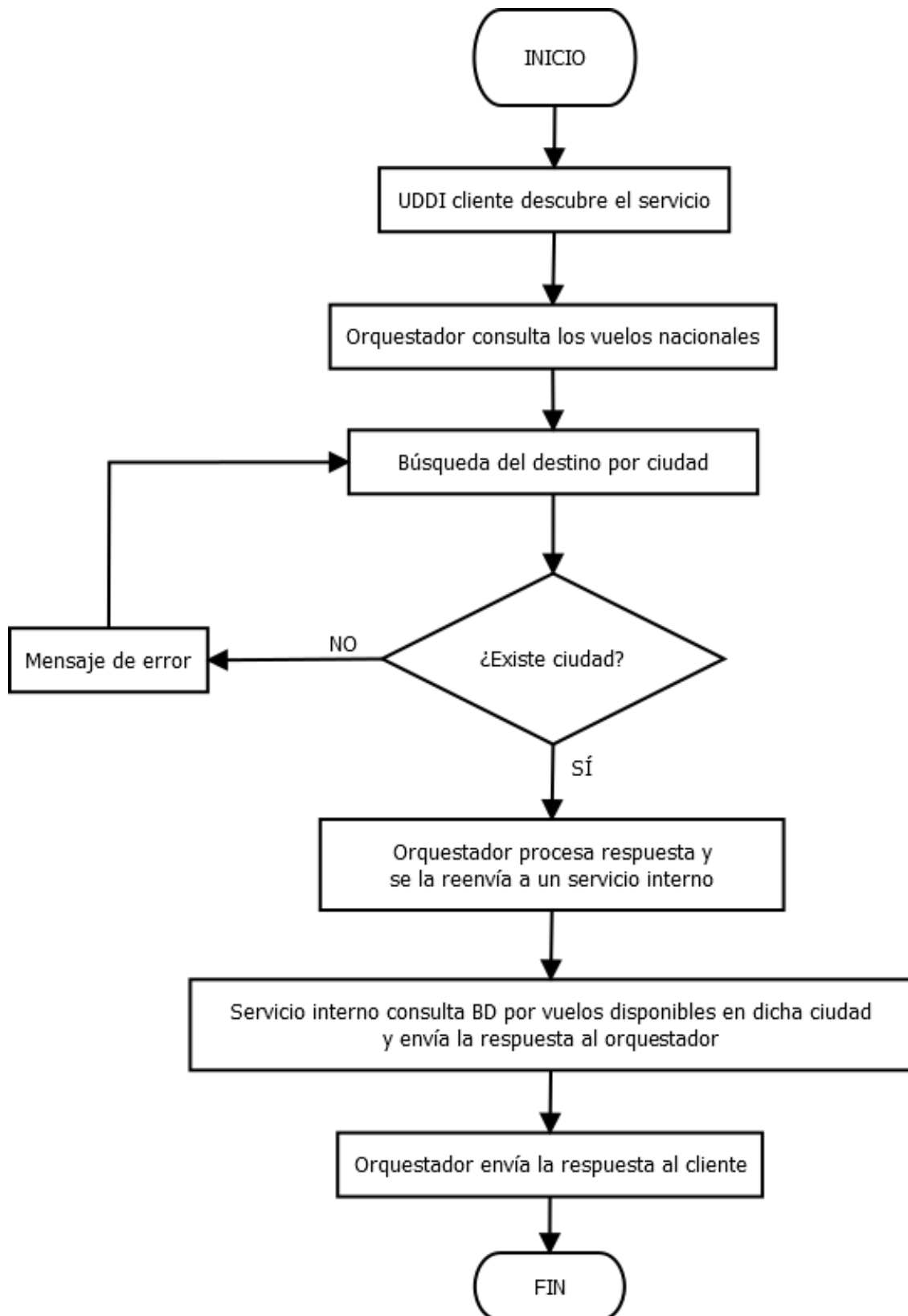
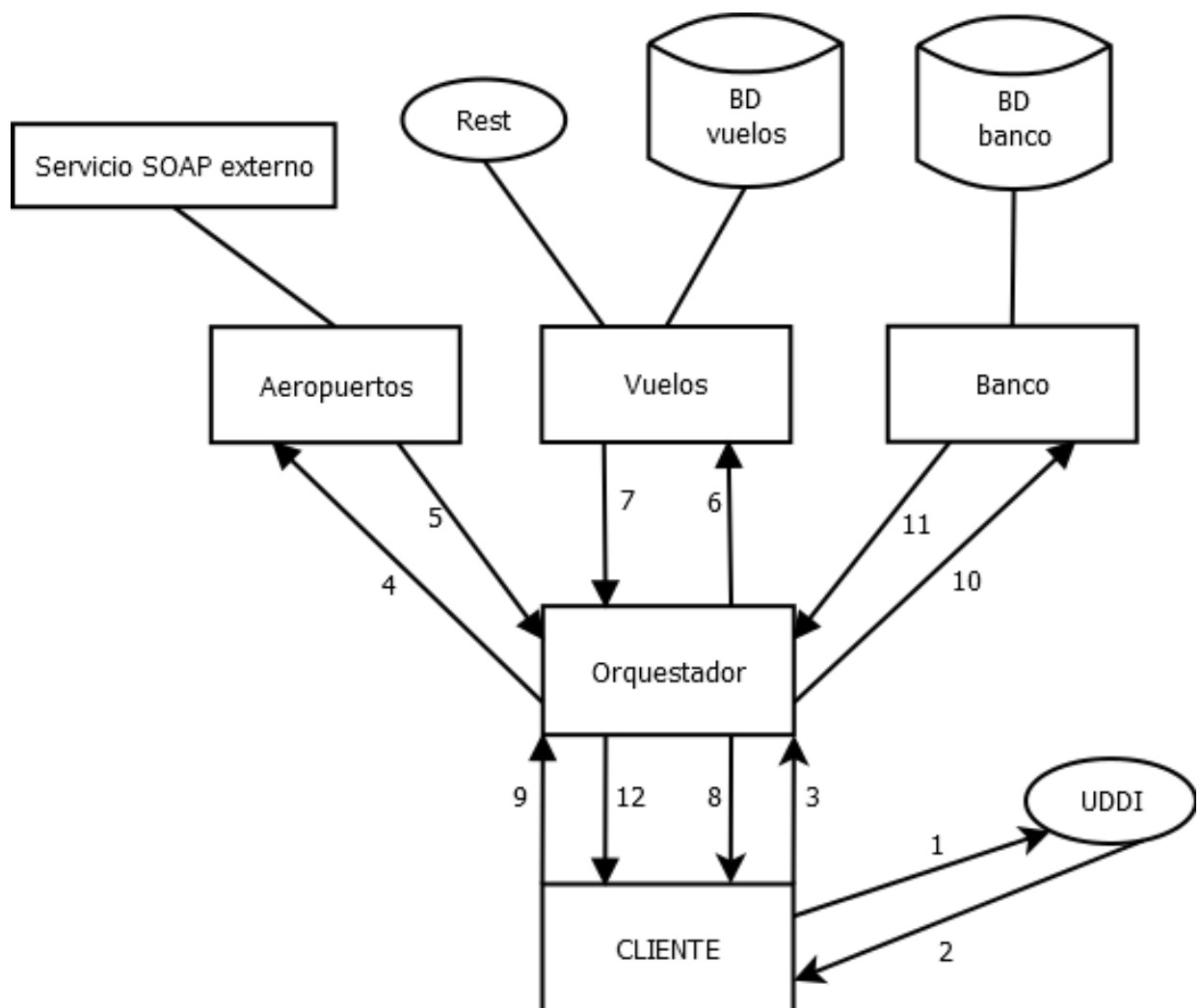


Diagrama de orquestación

Haremos uso de la orquestación, donde un proceso central (servicio Web) lleva el control del resto de los servicios Web implicados y coordina la ejecución de las diferentes operaciones sobre dichos servicios web. El coordinador central de la orquestación es consciente de la meta a conseguir (los servicios orquestados no conocen que están implicados en un proceso de composición), por lo que la orquestación se centraliza mediante definiciones explícitas de las operaciones y del orden en el que se deben invocar los servicios.



Funcionamiento

1, 2, 3 – El cliente descubre nuestro servicio mediante UDDI y realiza una búsqueda indicando la ciudad de origen y de destino.

4, 5 – El orquestador consulta al servicio interno “Aeropuertos”, el cual a su vez consulta un servicio externo y que devuelve el código del aeropuerto de origen y destino.

6, 7 – Si se ha recibido un error por parte de “Aeropuertos” se lo comunica al cliente. En caso contrario continua, iniciando una comunicación con el servicio interno “Vuelos”. “Vuelos” devuelve una lista de vuelos disponibles (fechas, horarios, precios,...) obtenidos por la consulta a su base de datos local o a una página web mediante el uso de REST.

8, 9 – El orquestador le muestra los resultados al cliente quién decidirá si seguir adelante y comprar un billete (eligiendo código de vuelo e identificándose para el pago) o no hacer nada más y salir.

10, 11 – Cuando el orquestador recibe información de pago y código de vuelo contacta con el servicio interno “Banco” para realizar el pago de forma asíncrona. En este servicio usamos handlers para gestión de errores (transferencias realizadas correctamente, identificación del comprador correcta,...).

12 – Al ser la comunicación con el banco asíncrona, para avisar al cliente del resultado se le enviará un email indicando que todo ha procedido correctamente o el error correspondiente.

Servicio externo

El servicio externo escogido es un web service que devuelve distinta información sobre un aeropuerto en particular. Dicho servicio consta de cuatro métodos:

- GetAirportInformationByCountry
- getAirportInformationByAirportCode
- getAirportInformationByCityOrAirportName
- getAirportInformationByISOCountryCode

Estos servicios devuelven el código del aeropuerto, nombre del aeropuerto, país, abreviatura del país, código del país, y otras características como la altitud, latitud, longitud o uso horario.

En nuestro caso, utilizaremos el primer método (GetAirportInformationByCountry) para encontrar todos los vuelos nacionales y gestionarlos.

Endpoint del servicio:

<http://www.webservicex.net/airport.asmx?WSDL>

Más información sobre el servicio en el enlace:

<http://www.webservicex.net/New/Home/ServiceDetail/20>

WSDL de los servicios internos

Para implementar nuestro servicio web, contamos con tres servicios internos cuyas descripciones WSDL adjuntamos:

- Aeropuerto.wsdl
- Banco.wsdl
- Vuelos.wsdl
- Orquestador.wsdl

Handlers:

Implementaremos un handler que mostrará un mensaje de error en el caso de que la transacción bancaria no se haya realizado satisfactoriamente.

Este handler se encontrará en el flujo de tipo IN del banco a orquestador y en un flujo de tipo OUT del orquestador al cliente.