5 Programación

Este capítulo sirve como guía de información de la programación de los entregables a desarrollar:

* Transacciones API Transportes (véase en el apartado 5.1)
* Integración (véase en el apartado 5.2)
* Formulario de pruebas (véase en el apartado 5.3)

Se adjuntan como anexo los ficheros correspondientes a estos tres paquetes de software desarrollados.

La solución de la integración contiene también el proyecto de tests unitarios (véase en el aparatado 5.4).

5.1 Transacciones de la API de Transportes

En el apartado 5.1.1 de este capítulo, se indica el lenguaje de programación y librerías utilizadas para desarrollar las nuevas transacciones de la API de Transportes tal y como se han representado en el diseño expuesto en el capítulo 4.1 de esta memoria, y en base a los requisitos funcionales y no funcionales enumerados en el capítulo 3.

A continuación, en el apartado 5.2 se cita el código programado y se enumeran los ficheros desarrollados que se adjuntarán como anexo a esta memoria.

En el apartado 5.3 se muestra el formulario de pruebas desarrollado y un ejemplo de uso. Finalmente, en el apartado 5.4 contenido dentro de este capítulo, se explican los escenarios de test desarrollados y su importancia para la mantenibilidad del código.

5.1.1 Lenguaje/tecnología

El lenguaje de programación con el que se programan las nuevas transacciones de la API de Transportes ya citadas en anteriores capítulos: DMR, RMR y EMR, así como los cambios necesarios introducidos en los elementos ya existentes en otras transacciones, es Visual Basic.

Se ha escogido este lenguaje porque el resto de transacciones que ya incluye la API está desarrollado con este mismo lenguaje. Sin embargo, una alternativa válida para desarrollar la misma API desde 0 en otro lenguaje puede ser perfectamente C#. Ofrece las mismas funcionalidades que ya se han utilizado para desarrollar la API de Transportes y además permite crear los ficheros necesarios para subirse en los frontales del servicio wsdl que existe en el sistema de TravelgateX.

Visual Basic (o C#) es necesario para que el cliente pueda atacar a las integraciones por .dll a través del entorno de test que tiene TravelgateX para que éste realice pruebas antes de subir los cambios de la API en el entorno de pre-producción o producción.

Al ser un lenguaje del paradigma de programación “Programación Orientada a Objetos” (POO), se construyen las clases necesarias para la creación de las nuevas llamadas. Todas las clases, elementos y atributos que se detallan en el apartado siguiente (5.1.2) son públicos.

Las tres llamadas desarrolladas son clases serializables, por lo tanto no se pueden heredar. Además las tres deben heredar de las clases padre TransportationBaseRQ para las peticiones y TransportationBaseRS para las respuestas.

Los elementos que incluyen las clases son del tipo XmlElementMultiLanguage y se indican los ElementName para cada idioma (español e inglés). Cuando un elemento tenga el mismo nombre para todos los idiomas con los que se vaya a poder utilizar la API, es importante definirlos con el tipo XmlElement.

Los atributos son del tipo XmlAttributeMultiLanguage y, al igual que los elementos, se indica el nombre del atributo en los dos idiomas.

Los elementos que sean raíz de una transacción xml deben definirse del tipo XmlRootMultiLanguage.

Para la programación de las clases, elementos y atributos, son suficientes las librerías del sistema (Microsoft) XML y librerías internas de la API de Transportes, ya que se van instanciar clases que deben ser heredadas por las nuevas peticiones y respuestas desarrolladas.

5.1.2 Descripción de las transacciones programadas

En este capítulo se detallan los elementos y atributos que contienen las nuevas llamadas de la API de Transportes que se usarán en la integración de Amadeus para modificar reservas. Los elementos y atributos expuestos en las tablas *Tabla 23*, *Tabla 24* y *Tabla 25*, no representan el código en si (programado en Visual Basic), sino la definición (cardinalidad y descripción) de cada uno de ellos. Éste formato es que se ha utilizado para documentar los cambios realizado en la API y que, como se ha establecido en el alcance del proyecto, es uno de los entregables que se han de presentar en TravelgateX.

En esta memoria se describe aproximadamente el 60% de los elementos que se han desarrollado para las nuevas transacciones DMR, RMR y EMR, los más importantes de cada transacción. El código de cada una de las transacciones se adjunta como anexo. (¿cómo pongo el anexo?). En el código, cada uno de los elementos cuenta con una descripción /\*comentario\*/.

Tabla 23: DMR

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento | Card. | Tipo | Significado |
| DisponibilidadModificarReservaRQ | 1..1 | obj. | Nodo raíz |
| @tipoViaje | 1..1 | enum | Tipo de viaje: (OW, RT, OJ o CT) |
| Localizador | 1..1 | obj. | Contiene información del localizador de la reserva |
| Localizador/@id | 1..1 | string | Identificador del localizador |
| Localizador/@tipoLocalizador | 1..1 | enum | Tipo de localizador: (PROVEEDOR, UNIVERSAL, EMISION, TRANSPORTISTA, REEMSOLSO, SERVICIO, CHECKIN) |
| Preferencias | 1..1 | obj. | Contiene las preferencias de la búsqueda |
| Preferencias/@claseCabina | 1..1 | enum | Clase cabina preferida para la búsqueda: (N, Y, C, F, CAMAROTE, YP) |
| Preferencias/@incluirLowCost | 1..1 | bool | Si es true, se buscarán vuelos de compañías lowcost |
| Preferencias/@soloDirectos | 1..1 | bool | Si es tru, se buscarán solo vuelos directos (sin escalas) |
| Preferencias/CompaniasConexion | 0..1 | obj. | Incluye una lista de compañías |
| Preferencias/CompaniasConexion  /CompaniaConexion | 1..n | obj. | Compañía incluida o excluida de la búsqueda |
| Preferencias/CompaniasConexion  /CompaniaConexion/@cia | 1..1 | string | Código IATA compañía |
| Preferencias/CompaniasConexion  /CompaniaConexion/@modo | 1..1 | enum | Modo de filtro: (INCLUDED, EXCLUDED) |
| TramosDisponibilidad | 1..1 | obj. | Incluye una lista de tramos |
| TramosDisponibilidad/TramoDisponibilidad | 1..n | obj. | Incluye la información de un tramo de la búsqueda |
| TramosDisponibilidad/TramoDisponibilidad  /@id | 1..1 | int | Identificador único del tramo |
| TramosDisponibilidad/TramoDisponibilidad  /@fechaSalida | 1..1 | date | Fecha de salida |
| TramosDisponibilidad/TramoDisponibilidad  /@fechaLlegada | 1..1 | date | Fecha de llegada |
| TramosDisponibilidad/TramoDisponibilidad  /@horaSalida | 0..1 | string | Hora de salida |
| TramosDisponibilidad/TramoDisponibilidad  /@horaLlegada | 0..1 | string | Hora de llegada |
| TramosDisponibilidad/TramoDisponibilidad  /@accion | 1..1 | enum | Modificación a realizar al tramo: (N, K, KF, R, C, A) |
| TramosDisponibilidad/TramoDisponibilidad  /LocOrigen | 1..1 | obj. | Incluye la información de la localización de partida del vuelo |
| TramosDisponibilidad/TramoDisponibilidad  /LocOrigen/@codigo | 1..1 | string | Código IATA del aeropuerto o ciudad |
| TramosDisponibilidad/TramoDisponibilidad  /LocOrigen/@radio | 0..1 | int | Radio de búsqueda respecto a la localización |
| TramosDisponibilidad/TramoDisponibilidad  /LocDestino | 1..1 | obj. | Incluye la información de la localización de destino del vuelo |
| DisponibilidadModificarReservaRS | 1..1 | obj. | Nodo raíz |
| Transportes | 1..1 | obj. | Incluye información de segmentos y tarifas obtenidos en la búsqueda |
| Transportes/@totalTarifas | 1..1 | int | Número de tarifas obtenidas en la búsqueda |
| Transportes/Tarifas | 1..1 | obj. | Contiene una lista de tarifas |
| Transportes/Tarifas/Tarifa | 1..n | obj. | Contiene información de la tarifa |
| Transportes/Tarifas/Tarifa/@id | 1..1 | int | Identificador único de tarifa |
| Transportes/Tarifas/Tarifa/@tipoTarifa | 1..1 | enum | Tipo de tarifa: (OW, RT, OJ o CT) |
| Transportes/Tarifas/Tarifa/@tieneObFees | 1..1 | bool | Si es true, la tarifa tiene cargos por pago con tarjeta |
| Transportes/Tarifas/Tarifa/Condiciones | 0..1 | obj. | Contiene una lista de condiciones |
| Transportes/Tarifas/Tarifa/Condiciones  /Condicion | 1..n | obj. | Condición de tarifa |
| Transportes/Tarifas/Tarifa/Condiciones  /Condicion/ConceptoTipificado | 1..1 | obj. | Información de la condición |
| Transportes/Tarifas/Tarifa/Opciones | 1..1 | obj. | Contiene una lista de opciones |
| Transportes/Tarifas/Tarifa/Opciones/Opcion | 1..n | obj. | Alternativas de tramos que incluye la tarifa |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /@id | 1..1 | int | Identificador único de opción |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /@refTramoDisponibilidad | 1..1 | int | Referencia al identificador único del tramoDisponibilidad de la petición DMR |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /@numEscalas | 1..1 | int | Cantidad de escalas del tramo |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /@placa | 1..1 | string | Compañía validadora de la reserva |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /@familiaTarifa | 0..1 | string | Familia de la tarifa que proporciona la compañía que vende el vuelo |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos | 1..1 | obj. | Contiene una lista de referencias segmento |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento | 1..n | obj. | Contiene la referencia al identificador único de un segmento de la lista de segmentos obtenidos en la búsqueda e información aplicada a dicho segmento |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /@refSegmento | 1..1 | int | Referencia al identificador único de segmento |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /ClasesSegmento | 1..1 | obj. | Contiene una lista de clasesSegmento |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /ClasesSegmento/ClaseSegmento | 1..n | obj. | Contiene información aplicada al segmento y a un pasajero (hay una claseSegmento por pasajero) |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /ClasesSegmento/ClaseSegmento  /@claseCabina | 1..1 | enum | Clase cabina del vuelo |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /ClasesSegmento/ClaseSegmento  /@clase | 1..1 | string | Clase del vuelo |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /ClasesSegmento/ClaseSegmento  /@refPasajero | 1..1 | int | Referencia al identificador úncio del pasajero (dentro de ConfiguracionPasjero) |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /ClasesSegmento/ClaseSegmento  /@fareBasis | 0..1 | string | Código de la tarifa aplicado por el GDS o compañía validadora |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /ClasesSegmento/ClaseSegmento  /@fareType | 1..1 | enum | Tipo de tarifa: (PUB, PRI, NEGO, CORP) |
| Transportes/Tarifas/Tarifa/Opciones/Opcion  /ReferenciasSegmentos/ReferenciaSegmento  /ClasesSegmento/ClaseSegmento  /@avail | 1..1 | int | Cantidad de asientos disponibles para la ClaseSegmento |
| Transportes/Tarifas/Tarifa/DesgloseImporte | 1..1 | obj. | Contiene información del precio de la tarifa |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /@moneda | 1..1 | string | Moneda de pago |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /@importeTotal | 1..1 | dec | Importe total de la tarifa |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /@importeNoComisionable | 1..1 | dec | Importe no comisionable de la tarifa |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /@comision | 1..1 | dec | Importe de la comisión |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /Cargos | 0..1 | obj. | Contiene una lista de cargos |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /Cargos/Cargo | 1..n | obj. | Importes extra no incluidos en el importe total de la tarifa |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /Cargos/Cargo/@tipo | 1..1 | string | Tipo de cargo (información del cargo) |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /Cargos/Cargo/@importe | 1..1 | dec | Importe del cargo |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /DesglosesPasajeros | 1..1 | obj. | Contiene una lista de DesglosePasajero |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /DesglosesPasajeros/DesglosePasajero | 1..n | obj. | Contiene los importes desglosados por tipo de pasajero |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /DesglosesPasajeros/DesglosePasajero  /@tipoPax | 1..1 | enum | Tipo de pasajero: (ADT, CHD, INF) |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /DesglosesPasajeros/DesglosePasajero  /@importe | 1..1 | dec | Importe del tipo de pasajero |
| Transportes/Tarifas/Tarifa/DesgloseImporte  /DesglosesPasajeros/DesglosePasajero  /@tasas | 1..1 | dec | Tasas del tipo de pasajero |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros | 1..1 | obj. | Contiene la lista de ConfiguracionPasajero |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  / ConfiguracionPasajeros | 1..n | obj. | Información del pasajero |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  /ConfiguracionPasajero/@id | 1..1 | int | Identificador único del pasajero |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  /ConfiguracionPasajero/@refPasajero | 1..1 | int | Referencia única del pasajero (se usa para referenciar las maletas incluidas por pasajero) |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  /ConfiguracionPasajero/@edad | 0..1 | int | Edad del pasajero |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  /ConfiguracionPasajero/@nacionalidad | 0..1 | string | Nacionalidad del pasajero |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  /ConfiguracionPasajero/@tipoPax | 1..1 | enum | Tipo del pasajero |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  /ConfiguracionPasajero/Bonificaciones | 0..1 | obj. | Bonificaciones aplicadas al pasajero |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  /ConfiguracionPasajero/Bonificaciones  /@residente | 0..1 | enum | Tipo de residente: (BP, BI, CE, RE…) |
| Transportes/Tarifas/Tarifa  /ConfiguracionesPasajeros  /ConfiguracionPasajero/Bonificaciones  /@familiaNumerosa | 0..1 | enum | Tipo de familia numerosa: (F1, F2) |
| Transportes/Segmentos | 1..1 | obj. | Contiene una lista de segmentos |
| Transportes/Segmentos/Segmento | 1..n | obj. | Contiene información del vuelo |
| Transportes/Segmentos/Segmento/@id | 1..1 | int | Identificador único del vuelo |
| Transportes/Segmentos/Segmento  /@numTransporte | 1..1 | string | Número de vuelo (p.e. IB98950) |
| Transportes/Segmentos/Segmento  /@operatingCia | 1..1 | string | Aerolínea que opera el vuelo |
| Transportes/Segmentos/Segmento  /@marketingCia | 1..1 | string | Aerolínea que vende el vuelo |
| Transportes/Segmentos/Segmento  /@terminalSalida | 0..1 | string | Terminal de salida |
| Transportes/Segmentos/Segmento  /@terminalLlegada | 0..1 | string | Terminal de llegada |
| Transportes/Segmentos/Segmento  /@fechaSalida | 1..1 | date | Fecha de salida |
| Transportes/Segmentos/Segmento  /@fechaLlegada | 1..1 | date | Fecha de llegada |
| Transportes/Segmentos/Segmento  /@tieneParadaTecnica | 1..1 | bool | Si es true, el vuelo tiene una o más paradas técnicas |
| Transportes/Segmentos/Segmento  /ParadaTecnicaDetalles | 0..1 | obj. | Detalles de las paradas técnicas |
| Transportes/Segmentos/Segmento  /LocOrigen | 1..1 | obj. | Contiene información de la localización de partida del vuelo |
| Transportes/Segmentos/Segmento  /LocDestino | 1..1 | obj. | Contiene información de la localización de llegada del vuelo |
| Transportes/Tickets | 1..1 | obj. | Contiene una lista de tickets (billetes) |
| Transportes/Tickets/Ticket | 1..n | obj. | Contiene información del ticket |
| Transportes/Tickets/Ticket/@id | 1..1 | int | Identificador único del ticket |
| Transportes/Tickets/Ticket/@numTicket | 1..1 | string | Número del ticket |
| Transportes/Tickets/Ticket/@nomPax | 0..1 | string | Nombre del pasajero asociado al ticket |
| Transportes/Tickets/Ticket/@tipoPax | 1..1 | enum | Tipo de pasajero asociado al ticket |
| Transportes/Tickets/Ticket/@tipoTicket | 1..1 | enum | Tipo de ticket: (eTicket, Extra) |
| Transportes/Tickets/Ticket/@estado | 1..1 | enum | Estado en el que se encuentra el ticket: (Open, Confirmed, Voided, Refunded) |
| Transportes/Localizadores | 1..n | obj. | Contiene información del localizador de la reserva |

Tabla 24: RMR

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento | Card. | Tipo | Significado |
| ReservaModificarReservaRQ | 1..1 | obj. | Nodo raíz |
| @deltaPrice | 1..1 | dec | Importe diferencial aceptado entre el precio de la tarifa obtenida en DMR y el precio que se obtenga en la RMR |
| Desgloses | 1..1 | obj. | Contiene una lista de desgloses |
| Desgloses/Desglose | 1..n | obj. | Contiene información de los segmentos y la tarifa |
| Desgloses/Desglose/@id | 1..1 | int | Identificador único del desglose |
| Desgloses/Desglose/@refTarifa | 1..1 | int | Referencia al identificador único de la tarifa de DMR |
| Desgloses/Desglose/@tieneObFees | 1..1 | bool | Si es true, la tarifa tiene cargos por pago con tarjeta |
| Desgloses/Desglose/@placa | 1..1 | string | Compañía validadora de la reserva |
| Desgloses/Desglose/Condiciones | 0..1 | obj. | Contiene una lista de condiciones |
| Desgloses/Desglose/Tramos | 1..1 | obj. | Contiene una lista de tramos |
| Desgloses/Desglose/Tramos/Tramo | 1..n | obj. | Contiene información de lo opción seleccionada en DMR |
| Desgloses/Desglose/Tramos/Tramo/@id | 1..1 | int | Identificador único de tramo |
| Desgloses/Desglose/Tramos/Tramo  /@familiaTarifa | 0..1 | string | Familia de la tarifa que proporciona la compañía que vende el vuelo |
| Desgloses/Desglose/Tramos/Tramo  /SegmentosTramo | 1..1 | obj. | Contiene una lista de segmentoTramo |
| Desgloses/Desglose/Tramos/Tramo  /SegmentosTramo/SegmentoTramo | 1..n | obj. | Contiene información de los vuelos y las clases asociadas a cada uno |
| Desgloses/Desglose/Tramos/Tramo  /SegmentosTramo/SegmentoTramo/@id | 1..1 | int | Identificador único de segmentoTramo |
| Desgloses/Desglose/Tramos/Tramo  /SegmentosTramo/SegmentoTramo  /Segmento | 1..1 | obj. | Contiene información del vuelo |
| Desgloses/Desglose/Tramos/Tramo  /ClasesSegmento | 1..1 | obj. | Contiene una lista de clases segmento |
| Transportes/Tarifas/Tarifa/DesgloseImporte | 1..1 | obj. | Contiene información del precio de la tarifa |
| Desgloses/Desglose/ConfiguracionesPasajero | 1..1 | obj. | Contiene una lista de configuraciones pasajero |
| InfoPago | 0..1 | obj. | Información de pago |
| InfoPago/@tipoPago | 1..1 | string | Tipo de pago: (CARD, CASH) |
| InfoPago/DatosPago | 0..1 |  | Contiene una lista de datos de pago |
| InfoPago/DatosPago/DatoPago | 1..n |  | Contiene información del método de pago |
| InfoPago/DatosPago/DatoPago/@numPlazos | 1..1 | int | Número de plazos para realizar el pago |
| InfoPago/DatosPago/DatoPago/Tarjeta | 1..1 |  | Información de la tarjeta |
| InfoPago/DatosPago/DatoPago/Tarjeta  /@tipoProv | 1..1 | string | Tipo de proveedor (p.e. Master Card o Visa) |
| InfoPago/DatosPago/DatoPago/Tarjeta  /@titular | 1..1 | string | Titular de la tarjeta |
| InfoPago/DatosPago/DatoPago/Tarjeta  /@numero | 1..1 | int | Número de la tarjeta |
| InfoPago/DatosPago/DatoPago/Tarjeta  /@cvc | 0..1 | int | Código de seguridad |
| InfoPago/DatosPago/DatoPago/Tarjeta  /@mesCaducidad | 1..1 | int | Mes de caducidad |
| InfoPago/DatosPago/DatoPago/Tarjeta  /@añoCaducidad | 1..1 | int | Año de caducidad |
| Tickets | 1..1 | obj. | Contiene una lista de tickets |
| Localizador | 1..1 | obj. | Contiene información del localizador de la reserva |
| ReservaModificarReservaRS | 1..1 | obj. | Nodo raíz |
| @tipoEmision | 1..1 | string | Tipo de emisión que ha de realizarse en EMR |
| Localizadores | 1..n | obj. | Contiene información del localizador de la reserva |
| Tickets | 1..1 | obj. | Contiene una lista de tickets |
| Pasajeros | 1..1 | obj. | Contiene una lista de pasajeros |
| Pasajeros/Pasajero | 1..n | obj. | Contiene información del pasajero |
| Pasajeros/Pasajero/@id | 1..1 | int | Identificador único del pasajero |
| Pasajeros/Pasajero/@tratamiento | 1..1 | enum | Tratamiento del pasajero (MR, MRS ,INF, CHD) |
| Pasajeros/Pasajero/@nombre | 1..1 | string | Nombre |
| Pasajeros/Pasajero/@apellidos | 1..1 | string | Apellidos |
| Pasajeros/Pasajero/@fechaNacimiento | 1..1 | date | Fecha de nacimiento |
| Pasajeros/Pasajero/@tipoDocumento | 0..1 | enum | Tipo de documentación de identificación: (NATIONAL\_ID, PASSPORT, RESIDENT\_ID, FOREIGN\_PASSPORT, BIRTH\_NOTIFICATION) |
| Pasajeros/Pasajero/@idDocumento | 0..1 | string | Número del documento de identificación |
| Pasajeros/Pasajero/@caducidadDocumento | 0..1 | date | Fecha de caducidad del documento de identificación |
| Pasajeros/Pasajero/@nacionalidad | 0..1 | string | Nacionalidad del pasajero |
| Pasajeros/Pasajero/@sexo | 0..1 | string | Sexo del pasajero |
| Pasajeros/Pasajero/ BonificacionesAplicadas | 0..1 | obj. | Bonificaciones aplicadas al pasajero |
| Pasajeros/Pasajero/BonificacionesAplicadas  /@municipioResidente | 0..1 | string | Municipio del residente |
| Pasajeros/Pasajero/BonificacionesAplicadas  /@codigoDocumentoFamiliaNumerosa | 0..1 | string | Documento de familia numerosa |
| Pasajeros/Pasajero/BonificacionesAplicadas  /@comunidadFamiliaNumerosa | 0..1 | string | Comunidad de familia numerosa |
| Pasajeros/Pasajero/BonificacionesAplicadas  /@codigoCertificadoResidente | 0..1 | string | Código de certificado de residente |
| Factura | 1..1 | obj. | Factura de la reserva |
| Factura/@compania | 1..1 | string | Compañía validadora de la reserva |
| Factura/@lastTicketingDate | 1..1 | date | Fecha del último checkin |
| Factura/DesgloseImporte | 1..1 | obj. | Contiene información del precio de la tarifa |

Tabla 25: EMR

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento | Card. | Tipo | Significado |
| EmitirModificarReservaRQ | 1..1 | obj. | Nodo raíz |
| @tipoEmision | 1..1 | string | Tipo de emisión que ha de realizarse en EMR |
| @localizador | 1..1 | string | Localizador de la reserva |
| Tickets | 1..1 | obj. | Contiene una lista de billetes |
| EmitirModificarReservaRS | 1..1 | obj. | Nodo raíz |
| Tickets | 1..1 | obj. | Contiene una lista de billetes |
| Localizadores | 1..n | obj. | Contiene información del localizador de la reserva |

5.2 Integración

En este capítulo se explica el lenguaje y la tecnología utilizada para programar la integración y se enumeran los ficheros desarrollados.

5.2.1 Lenguaje/tecnología

El lenguaje de programación con el que se programa la integración es Visual Basic. Se ha escogido este lenguaje porque la integración ya estaba desarrollada en este lenguaje.

Visual Basic (o C#) es necesario para que el cliente pueda atacar a las integraciones por .dll a través del entorno de test que tiene TravelgateX para que éste realice pruebas antes de subir los cambios de la integración en el entorno de pre-producción o producción.

5.2.2 Funcionalidades destacadas

TimeoutManager

Es la funcionalidad que controla el tiempo de ejecución de la transacción. Se declara indicando el tiempo de corte (timeout) especificado en el atributo @timeout de la petición.

GetRemainingMilliseconds

Devuelve el tiempo que falta para llegar al timeout.

GetEllapsedMilliseconds

Devuelve el tiempo que ha pasado desde que ha empezado la ejecución de la transacción.

ConnectionManager

Es la funcionalidad que se encarga de crear el cliente SOAP y utilizar dicho cliente para realizar la petición al proveedor. Contiene todas las llamadas de Amadeus desarrolladas en la integración. Se han desarrollado derivados de esta funcionalidad para para paquete de Web Services de Amadeus (p.e. ConnectionManagerBook para todas las llamadas del paquete de Web Services de tarificación y reserva de recomendaciones de Amadeus).

BookingRetrieve

Clase que monta toda la información que se recibe al recuperar la reserva.

Eligibility

Clase que se encarga de construir la petición para la llamada Ticket\_CheckEligibility y procesar y parsear la respuesta recibida. Si un ticket no es modificable, se añadiría un error de “Ticket no modificable” en la respuesta de DMR.

Travelboard

Clase que se encarga de construir la petición para la llamada Ticket\_ATCShopperMasterPricerTravelBoardSearch y procesar y parsear la respuesta recibida. Monta todas las tarifas y los segmentos a partir de los datos obtenidos en la respuesta.

BookingMake

Clase que monta toda la información para realizar la reserva de segmentos y pasajeros. Además se utiliza para montar las facturas de la misma a partir de los datos obtenidos

RepricePNR

Clase que se encarga de construir la petición para la llamada Ticket\_RepricePNRWithBookingClass y procesar y parsear la respuesta recibida.

ReissueConfirmedPricing

Clase que se encarga de construir la petición para la llamada Ticket\_ReissueConfirmedPricingTicketInfoy procesar y parsear la respuesta recibida.

GetTipoEmision

Funcionalidad que calcula el tipo de emisión que ha de realizarse en la transacción EMR a partir de:

* repriceType: tipo de reprice que Amadeus indica
* isThereEMD: Si el PNR tiene cargos EMD o no
* tipoEMD: tipo de EMD que hay en el PNR
* formOfPayment: forma de pago con el que se abone el importe de la modificación de la reserva
* addictionalCollection: la modificación tiene importe adicional

Se obtendrá uno de los siguientes tipos de emisión:

* Revalidate without EMD
* Revalidate with EMD without additional collection
* Reissue without EMD with additional collection
* Reissue with EMD only penalti
* Reissue with EMD only residual value without additional collection
* Reissue with EMD only residual value
* Reissue with EMD with penalty and residual value
* Reissue with EMD with penalty and residual value without additional collection

DoEndTransactionAndRetrieve

Confirma los cambios realizados en el PNR y lo reabre.

DoIgnoreAndRetrieve

Ignora los cambios realizados en el PNR y lo reabre.

ATC\_IssueTicket

Realiza la revalidación o reemisión de los billetes dependiendo del tipo de emisión obtenida del atributo @tipoEmision de la petición EMR.

ATC\_IssueCombined

Realiza la reemisión de los billetes dependiendo de si la modificación de RMR ha generado un valor residual o una penalización por modificación.

5.2.3 Integración (ficheros)

En principio aquí solo tengo que poner el anexo al código.

5.3 Formulario de pruebas

En el sub-apartado 5.3.1 se describe el lenguaje y tecnología utilizada para programar el formulario de pruebas que se muestra con un ejemplo de uso a través de un vídeo en el apartado 5.3.2.

5.3.1 Lenguaje/tecnología

El lenguaje de programación con el que se programa el formulario de pruebas es Visual Basic. Se ha escogido este lenguaje por dos razones: es el lenguaje que tenemos que utilizar en las transacciones de la API y en la integración y consecuentemente se tiene más práctica y soltura. Otra razón de peso es la ligera mayor velocidad con la que se programa al carecer de puntos y coma, se tienen que utilizar menos paréntesis y apenas se necesitan abrir o cerrar llaves.

5.3.2 Ejemplo de uso (vídeo)

Anexo al vídeo

5.4 Programación de tests unitarios

En este apartado se destaca la importancia de crear tests unitarios para la integración (5.4.1) y qué lenguaje de programación es el elegido para programar dichos tests. En el sub-apartado 5.4.2 también se enumeran y se describen algunas de las funcionalidades que proporcionan algunas librerías del lenguaje seleccionado.

5.4.1 Importancia

En el proyecto de test, se incluyen tests que comprueban el funcionamiento de transacciones o funcionalidades unitarias dentro de la integración. Casi todas las transacciones que tiene el GDS tienen su clase de test con uno o más tests unitarios.

Estos tests no realizan la llamada al proveedor en ningún entorno, sino que es el desarrollador el que especifica las transacciones que quiere testear. Así, por ejemplo, si se quiere comprobar que en unas determinadas condiciones, una reserva funciona correctamente, se puede hacer que el test se ejecute simulando el comportamiento real de la reserva, pero sin realizar la transacción. Esto se consigue sobrescribiendo las funciones que contactan con el proveedor asignándose siempre la misma respuesta (el string). Es muy útil, por ejemplo, cuando una reserva ha fallado y no se puede reproducir en producción. Al pasarle al test, las trazas del error, se reproducirá el comportamiento de la integración en las mismas condiciones pero sin hacer la llamada de verdad.

Cada test tiene al menos un string con la respuesta de alguna de las llamadas de la transacción, lo que conlleva que una clase de test puede llegar a tener una enorme cantidad de texto que llega incluso a ralentizar el PC con el que el desarrollador ejecute los tests. Este es uno de los motivos por los cuales es conveniente seguir un formato estándar al crear cada test. Otra razón de peso es la fácil legibilidad que implica que todos los tests en cualquier GDS sigan el mismo formato.

Pese a que los tests ayudan a que no se fastidie la integración al mantenerla, éstos también conllevan en ocasiones un mantenimiento. Ya que gran parte de los tests comprueban el funcionamiento de una transacción completa, cuando el flujo de una transacción cambia, entonces los tests también tendrán que cambiarse. Hay muchas otras causas que pueden provocar que tenga que actualizarse un test. Pese a todo, cuando un test falla, lo más probable es que se deba a algún cambio en la integración que el desarrollador haya hecho mal.

5.4.2 Lenguaje/tecnología

El lenguaje de programación con el que se programa el formulario de pruebas es C#.

Moq

Librería que permite la utilización de “mocks”, que no son más que objetos simulados que imitan el comportamiento de objetos reales. Se pueden comportamientos dependiendo de los parámetros que se le pasen al objeto “mock” e incluso ejecutar acciones complejas accediendo al mismo parámetro proporcionado al método simulado.

Ejemplo de uso en la integración:

* Creación de un mock ConnectionManager
* Setup de las llamadas de ConnectionManager para que devuelvan el objeto que deseemos (p.e. callPNR\_Retrieve que devuelva el PNR\_Reply que se indique en el return).
* Creación de un mock DMR para llamar al método process que ejecuta la transacción.