Ejercicio "TusLibros.com"

La editorial "TusLibros" desea actualizar su oferta de ventas de libros abriendo la posibilidad de hacer compras de manera on-line. El sistema que tienen actualmente funciona de manera batch, de tal manera que reciben archivos con los pedidos y devuelven archivos con los resultados de las compras. Su plan de actualización consta de dos etapas:

1) Desarrollar un sistema que reciba pedidos via una API Rest y dialogue de manera on-line con el Merchant Processor (validador de tarjetas)
2) Reemplazar el sistema actual haciendo que el nuevo sistema lea y genere los archivos que actualmente se usan como interfaz batch.

Como el Merchant Processor cobra por transacción, se debe validar la mayor cantidad de información posible antes de enviarla al mismo.
El Merchant processor ofrece un ambiente de desarrollo, pero también cobra por prueba, por lo que se debe evitar al máximo hacer pruebas con el mismo al menos que sea necesario (o sea, solo usar el ambiente de desarrollo del merchant processor luego de la integración).
El sistema debe guardar todas las compras realizadas para luego poder reponer stock, sacar estadísticas de ventas, etc.

El Merchant Processor provee una interface Rest para realizar los débitos a las tarjetas. La misma se ejecuta haciendo post a https://merchant.com/debit (para producción) y https://merchanttest.com/debit (para desarrollo) y recibe los siguientes parámetros:

1) creditCardNumber: Número de tarjeta de crédito
2) creditCardExpiration: Fecha de expiración con 2 dígitos para el mes y 4 para el año
3) creditCardOwner: Nombre del dueño de la tarjeta. Máximo de 30 caracteres
4) transactionAmout: Cantidad de la transacción, con un máximo de 15 dígitos para la parte entera y dos dígitos para la decimal (siempre se debe generar la parte decimal) utilizando el punto como separador.

Por ejemplo:
https://merchant.com/debit?creditCardNumber=5400000000000001&creditCardExpiration=072011&creditCardOwner=PEPE%20SANCHEZ&transactionAmount=123.50
Estos datos de ejemplo son los utilizados como datos válidos por el ambiente de desarrollo del Merchant Processor.

Si algún parámetro no tiene el formato correcto devuelve como código de HTTP 400 (Bad request), en caso contrario devuelve como resultado HTTP 200 con el siguiente contenido:

1) En caso exitoso: 0|OK
2) En caso de no poder realizar el débito: 1|DESCRIPCION_DE_ERROR
   En caso de que la transacción no se pudo realizar se desea pasar la descripción del error al usuario de sistema.

Lamentablemente el up-time del Merchant Processor no es muy bueno, por lo que si el mismo se encuentra caído cuando se desea validar una transacción, se debe generar el pedido en un archivo de input cuyo ID de cliente del nombre será TUSLIBROS (ver más abajo especificación de este archivo)

La interfaz Rest que ofrecerá el sistema debe permitir crear un carrito (el cual será válido durante 30 minutos luego de la última vez que se realizó alguna operación con él), agregar un libro con su cantidad al carrito ya creado, consultar el contenido de un carrito, hacer el check out y listar las compras de un cliente. Las interfaces son:

1) Recurso: /createCart
   Parámetros:
   clientId: ID del cliente que está creando el carrito
   password: Password del cliente que valida que puede operar con TusLibros.com
   Output:
   En caso de éxito: 0|ID_DEL_CARRITO
   En caso de error: 1|DESCRIPCION_DE_ERROR
2) Recurso: /addToCart
   Parámetros:
   cartId: Id del carrito creado con /createCart
   bookIsbn: ISBN del libro que se desea agregar. Debe ser un ISBN de la editorial
   bookQuantity: Cantidad de libros que se desean agregar. Debe ser >= 1.
   Output:
   En caso de éxito: 0|OK

En caso de error: 1|DESCRIPCION_DE_ERROR

3) Recurso: /listCart
   Parámetros:
   cartId: Id del carrito creado con /createCart
   Output:
   En caso de éxito: 0|ISBN_1|QUANTITY_1|ISBN_2|QUANTITY_2|....|ISBN_N|QUANTITY_N
   En caso de error: 1|DESCRIPCION_DE_ERROR
4) Recurso: /checkOutCart
   Parámetros:
   cartId: Id del carrito creado con /createCart
   ccn: Número de tarjeta de credito
   cced: Fecha de expiración con 2 digitos para el mes y 4 para el año
   cco: Nombre del dueño de la tarjeta.
   Output:
   En caso de éxito: 0|TRANSACTION_ID
   En caso de error: 1|DESCRIPCION_DE_ERROR
5) Recurso: /listPurchases
   Parámetros:
   clientId: ID del cliente que quiere ver que compras hizo
   password: Password del cliente que valida que puede operar con TusLibros.com
   Output:
   En caso de éxito: 0|ISBN_1|QUANTITY_1|....|ISBN_N|QUANTITY_N|TOTAL_AMOUNT
   En caso de error: 1|DESCRIPCION_DE_ERROR

Si el request realizado no cumple con las reglas sintácticas, se debe devolver como HTTP status el código 400 (Bad request). Si cumple con la sintaxis se debe devolver como HTTP status el código 200 (OK).

Los archivos utilizados como interface batch son:
De entrada: CLIENTE_INPUT_AAAA_MM_DD.csv con el siguiente formato: TipoDeRegistro,RestoDelRegistro donde:

1) Para el tipo de registro 1: Agregar un libro al carrito, en formato ISBN,QUANTITY (Crear el carrito si el mismo no existe)
2) Para el tipo de registro 2: Realizar el checkout del carrito con el siguiente formato: numero_de_tarjeta,fecha_de_expiración,nombre_del_dueño.

De Salida: CLIENTE_OUTPUT_AAAA_MM_DD.csv donde cada registro tiene el siguiente formato:

1) Resultado de Transacción: 0 para exito, 1 en caso de error
2) En caso de éxito, id_de_transacción, total_de_transacción
   En caso de error: descripción_del_error

Ejemplo:
Archivo TEMATIKA_INPUT_2010_02_01.csv
1,0321146530,2
1,1933988274,1
2,5400000000000001,072011,PEPE SANCHEZ
1,1933988274,3
2,5400000000000002,132012,KENT BECK

Archivo TEMATIKA_OUTPUT_2010_02_01.csv
1,10533,60.53
2,INVALID EXPIRATION DATE