

SERVICIOS

- [LISTA DE SERVICIOS](#)

LISTA DE SERVICIOS

Dominio: <https://sysqa.shalombus.pe/#/>

SERVICIOS:

1. Listar orígenes:

ruta: api/listar-origen
Method: POST
body: {}

Función: Al ingresar a la página web se ejecuta el servicio para listar todos los orígenes de shalom express habilitados.

2. Listar destinos:

ruta: api/listar-destino
Method: POST
body: { "ciudad": "1" }

Función: Al seleccionar un origen, listará todos los destinos que hay en shalom express habilitados.

3. Listar salidas:

ruta: api/listar-salidas
Method: POST

```
body:
{
  "idciudad": "1",
  "idterminal": "22",
  "idciudadDest": "2",
  "idterminalDest": "32",
  "fechaSalida": "2025-10-22",
  "fechaRegreso": "",
  "horaActual": "18:32:27"
}
```

Función: Al seleccionar un destino y origen, y llenar el campo de fecha de ida o retorno (opcional), listará todas las salidas habilitadas en web.

4. Listar asientos del bus:

ruta: api/listar-buses

Method: POST

```
body:
{
  "idsalida": "42391",
  "indicador": "0",
  "indice": "0g",
  "codigo": "HBGFT",
  "tipo": "salida"
}
```

Función: Al seleccionar una salida, listará los asientos del bus de primer piso como del segundo piso.

5. Bloquear y desbloquear asientos:

ruta: api/bloquear-asiento

Method: POST

```
body:
{
  "id": "42391",
  "asiento": "5",
  "user": "100000010",
  "codigo": "HBGFT"
}
```

Función: Bloquear y desbloquear asiento(s)

6. Buscar salidas disponibles(validacion por back):

ruta: api/buscar-salida
Method: POST
body: { "id": "42391" }

Función: Busca la salida mediante su ID y compara con el día y hora actual si su fecha de salida aun no vence. en caso venza la redirigira para que elija nuevamente otra salida.

7. Validar la fecha y hora de bloqueo y precio actualizado:

ruta: api/unlock-asiento-validate
Method: POST
body: { "asiento": ",5", "salida": "42391" }

Función: Valida si la fecha y hora de bloqueo es menor a 10 min entonces procedera con el pago del viaje seleccionado. Pero si es mayor devolvera con un "no" como mensaje de respuesta y ya no procederá con el proceso.

8. Verificar asiento vendido o bloqueado:

ruta: api/verificar-asiento
Method: POST
body: asientos[0][salida] : 42392 asientos[0][asiento] : 2 asientos[0][codeventa] : '972M5'

Función: verifica si el asiento que el pasajero ha elegido de esa salida ya ha sido pagado o ocupado por otra persona, esa validación es interna, para no generar duplicados de comprobantes ni errores.

9. Consultar pasajero:

ruta: api/consulta-pasajero

Method: POST

body:

```
{  
  "codigo": "75844645"  
}
```

Función: Verifica que en la BD exista el pasajero, en caso no exista el cliente va a tener que llenar manualmente sus datos para que despues pueda guardarse en la bd.

10. Consultar pasajero:

ruta: api/monto-asientos

Method: POST

```
body:
{
  "asiento": " , 3",
  "salida": "42420",
  "pasajeros": {
    "nombres": [
      "GIAN PIERO"
    ],
    "apellidos": [
      "VILLANUEVA GASTELLO"
    ],
    "dni": [
      "75013406"
    ],
    "tipodocu": [
      "DNI",
      "DNI",
      "DNI",
      "DNI"
    ],
    "genero": [
      "M",
      "M",
      "M",
      "M"
    ],
    "fecha": [
      "31/12/1969"
    ],
    "correo": [
      "pierovillanueva.g@hotmail.com"
    ],
    "celular": [
      "994289949"
    ],
    "ruc": [],
    "razon": [],
    "factura": []
  },
  "tarjeta": {
    "nombres": "Marco Antonio Bohorquez",
    "correo": "antoniob@overskull.pe",
    "tarjeta": "",
    "mes": "",
    "year": "",
    "ccv": "",
    "dni": "75013406",
    "apellidos": "",
    "celular": "",
    "terminos": true,
    "tipoDocumentoTar": "DNI"
  },
  "viajes": {
    "precio": 2,
    "salida": "42420",
    "llegada": "TER.LOS ANDES-HUANCAYO",
    "fecha": "25-11-2025",
    "servicio": "SOFA PREMIUM",
    "asientos": " , 3",
    "headtitleOrigen": "PLAZA NORTE - LIMA",
    "headTerminalOrigen": "PLAZA NORTE - LIMA",
    "headDireccionOrigen": "GRAN TERMINAL TERRESTE PLAZA NORTE",
```

Función: Valida enviando la salida o salidas involucradas ya sea solo ida o ida y retorno y retorna un monto que es el monto total de la salida o salidas.

11. Enviar info a culqi:

ruta: api/culqi-ajax
Method: POST
body: { "token": "tkn_test_43R3caryNsveGE61", "monto": 200, "email": "review@culqi.com", "cliente": "75013406", "celular": "", "nombre": "Marco Antonio Bohorquez", "apellido": "", "device_finger_print_id": null, "pasajeros": ["75013406"], "salidas": [{ "salida": "42420", "asiento": "3" }], "force_3ds": 0, "test": 1 }

Función: Con esta informacion, culqi valida que la transaccion se haya cobrado exitosamente de la tarjeta y del banco. Aqui devuelve en caso sea fraude o no una autentificacion mas, pero eso ya es un proceso interno de culqi.

12. Enviar info a culqi con auth 3ds:

ruta: api/culqi-ajax
Method: POST

```
body:
{
  "token": "tkn_test_43R3caryNsveGE61",
  "monto": 200,
  "email": "review@culqi.com",
  "cliente": "75013406",
  "celular": "",
  "nombre": "Marco Antonio Bohorquez",
  "apellido": "",
  "device_finger_print_id": "",
  "pasajeros": [
    "75013406"
  ],
  "salidas": [
    {
      "salida": "42420",
      "asiento": "3"
    }
  ],
  "force_3ds": 1,
  "eci": "05",
  "xid": "AAIBBYNoEwAAACcKhAJkdQAAAAA=",
  "cavv": "AAIBBYNoEwAAACcKhAJkdQAAAAA=",
  "protocolVersion": "2.2.0",
  "directoryServerTransactionId": "f8d17cc2-4d31-48a2-8dd8-e5b4cce3dfa5"
}
```

Función: Con esta informacion, ya culqi valida que no eres fraude y pasara tu transaccion normalmente.

13. Insertar o actualizar pasajero:

ruta: api/insertar-pasajero

Method: POST

body:

```
config[0]:dni
config[1]:insert
dato[0][tipoDocumento]:DNI
dato[0][documento]:75013406
dato[0][nombres]:GIAN PIERO
dato[0][apellidos]:VILLANUEVA GASTELLO
dato[0][sexo]:M
dato[0][correo]:pierovillanueva.g@hotmail.com
dato[0][telefono]:994289949
dato[0][fechaNacimiento]:1969-12-31
dato[0][asiento]:3
dato[0][documento_tar]:75013406
dato[0][nombres_tar]:Marco Antonio Bohorquez
dato[0][apellidos_tar]
compras:1
```

Función: Actualiza o inserta al pasajero para que se proceda con la venta del boleto y lo genere.

14. Genera un registro de pago antes de enviar a culqi:

ruta: api/generar-previo-pago

Method: POST

body:

```
object = charge
id = chr_test_CoNbpI5Nfks3j...
creation_date = 1764105869094
amount = 200
amount_refunded = 0
current_amount = 200
installments = 0
installments_amount = 0
currency_code = PEN
email = review%40culqi.com
description = Venta+de+Pasaje+Shalo...
source[object] = token
source[id] = tkn_test_43R3caryNsve...
source[type] = card
source[creation_date] = 1764105668988
source[email] = review%40culqi.com
source[card_number] = 445653*****1096
source[last_four] = 1096
source[active] = true
source[bin] = 445653
source[bin][issuer] = Bank+of+Fiction
source[bin][issuer_country] = United+States
source[bin][issuer_website] = https%3A%2F%2Fwwwr...
source[bin][issuer_country_code] = US
source[bin][issuer_phone] = %2B+(1)+937-859-6260
source[bin][object] = iin
source[bin][card_type] = Classic
source[bin][card_brand] = Visa
source[bin][card_level] = Gold
source[client][ip] = 10.230.11.245
source[client][device] = Chrome
source[client][country] = Peru
source[client][device_type] = escritorio
source[client][country_code] = PE
source[client][device_fingerprint] = 47adb8290b8954f6b40...
source[metadata] = 0
source[token_charge_reference] = chr_test_CoNbpI5Nfks3j...
outcome[type] = venta_exitosa
outcome[code] = AUTO0000
outcome[merchant_message] = La+operaci%C3%B3n+d...
outcome[user_message] = Su+compra+ha+sido+ex...
fraud_score = 0.5
antifraud_details[object] = client
antifraud_details[country_code] = PE
antifraud_details[first_name] = Marco+Antonio+Bohorq...
antifraud_details[last_name] = first_last_name
antifraud_details[address] =
antifraud_details[address_city] =
antifraud_details[phone] =
dispute = false
capture = true
partial = false
capture_date = 1764105869094
reference_code = 7000000038176
authorization_code = 046126
duplicated = false
metadata[cliente] = 75013406
metadata[nombre_cliente] = Marco+Antonio+Bohorq...
metadata[salidas] = 3
metadata[pasajeros] = 75013406
```

Función: inserta un registro de pago antes de mandarlo a culqi, cuando culqi responda con éxito o error ahí responderá con un estado y se validará el pago si ha sido procesado correctamente o no.

15. Crear registro de viaje asociado a boleto:

ruta: api/pagar-viaje
Method: POST
body: config:insert codeCulqi:BRWSLR compras:1 dato[0][ID_PASAJERO]:75013406 dato[0][NOMBRE_PASAJERO]:GIAN PIERO dato[0][APELLIDO_PASAJERO]:VILLANUEVA GASTELLO dato[0][ID_SUBIDA]:22 dato[0][ID_BAJADA]:32 dato[0][PASAJE]:2 dato[0][ADICIONAL]:0 dato[0][TIPOPAGO]:Deposito dato[0][NROOPERA]:700000038176 dato[0][ASIENTO_ESTADO]:v dato[0][ID_SALIDA]:42420 dato[0][ASIENTO_NRO]:3 dato[0][ID_USERCREA]:100000010 dato[0][ID_TERMVENTA]:38 dato[0][TIPOPAGO_ADICIONAL] dato[0][TIPOPAGO_PENALIDAD] dato[0][PENALIDAD]:0 dato[0][FH_CREACION] dato[0][TAR_NOMBRE]:Marco Antonio Bohorquez dato[0][TAR_APELLIDO] dato[0][TAR_DNI]:75013406 dato[0][CORREO]:pierovillanueva.g@hotmail.com dato[0][HORA_ESCALA]:22:00 dato[0][TELEFONO]:994289949 dato[0][TIPO_VIAJE]:ida codigo:E11NR

Función: genera el viaje con su boleto asociado.

16. Enviar correo electrónico:

ruta: api/envio-correoventa

Method: POST

body:

nombre:GIAN PIERO VILLANUEVA GASTELLO

codigo:E11NR

email:pierovillanueva.g@hotmail.com

asiento:3

origen:PLAZA NORTE - LIMA

destino:TERMINAL LOS ANDES

fecha:25-11-2025

hora:22:00

precio:2

operacion:700000038176

electronico:<https://www.nubefact.com/cpe/aa966efe-eb65-4da5-a712-8872ca8b00c2.pdf>

Función: Envía el correo al cliente quien hizo la compra.