

Guía de Integración API Relier

Cliente: HAZ Group **Versión del Documento:** 1.0 **Fecha:** 13 de Octubre, 2025

Estado: Listo para Producción

Tabla de Contenidos

1. [Descripción General](#)
 2. [Autenticación](#)
 3. [Endpoints de la API](#)
 4. [Formato de Solicitud/Respuesta](#)
 5. [Manejo de Errores](#)
 6. [Límites de Tasa y Seguridad](#)
 7. [Guía de Pruebas](#)
 8. [Ejemplos de Código](#)
 9. [Soporte](#)
-

Descripción General

Relier proporciona una API RESTful para recargas de tiempo aire móvil en México. Este documento describe el proceso de integración para que HAZ Group se conecte a la plataforma Relier.

Características de la API

- **Recargas en tiempo real** a los principales operadores mexicanos (Telcel, Movistar, AT&T, Unefon, Virgin Mobile)
- **Facturación en USD** con conversión automática a MXN a tipos de cambio competitivos
- **Seguimiento de transacciones** con IDs únicos para reconciliación
- **Alta disponibilidad** con SLA de 99.9% de tiempo de actividad
- **Autenticación segura** con claves API
- **Limitación de tasa** para prevenir abusos

URL Base

Producción:

```
https://latcom-fix-production.up.railway.app
```

Autenticación

Todas las solicitudes de API requieren autenticación mediante encabezados HTTP.

Encabezados Requeridos

```
Content-Type: application/json  
x-api-key: haz_prod_2025  
x-customer-id: HAZ_001
```

Notas de Seguridad

- Mantenga sus credenciales de API seguras

- Nunca confirme credenciales en el control de versiones
 - Use variables de entorno para el almacenamiento de credenciales
 - Rote las claves periódicamente (contacte a soporte para rotación de claves)
-

Endpoints de la API

1. Verificación de Salud

Verifica la disponibilidad de la API y el estado del sistema.

Endpoint: `GET /health` **Autenticación:** No requerida

Respuesta:

```
{
  "status": "OK",
  "timestamp": "2025-10-13T10:00:00.000Z",
  "uptime": 86400,
  "mode": "PRODUCTION",
  "services": {
    "database": {
      "connected": true,
      "status": "healthy"
    },
    "redis": {
      "connected": true,
      "status": "healthy"
    }
  }
}
```

2. Consultar Saldo

Recupera el saldo actual de tu cuenta y el límite de crédito.

Endpoint: `GET /api/balance` **Autenticación:** Requerida

Ejemplo de Solicitud:

```
curl -X GET https://latcom-fix-production.up.railway.app/api/balance \
-H "x-api-key: haz_prod_2025" \
-H "x-customer-id: HAZ_001"
```

Respuesta:

```
{
  "success": true,
  "customer_id": "HAZ_001",
  "company_name": "HAZ Group",
  "current_balance": 9998.92,
  "credit_limit": 10000,
  "currency": "USD"
}
```

3. Procesar Recarga (Síncrona)

Procesa una recarga móvil y espera el resultado.

Endpoint: `POST /api/enviadespensa/topup` **Autenticación:** Requerida **Límite de Tasa:** 200 solicitudes por minuto

Cuerpo de la Solicitud:

```
{  
  "phone": "5566374683",  
  "amount": 20,  
  "reference": "ORDEN_12345"  
}
```

Descripción de Campos:

Campo	Tipo	Requerido	Descripción
phone	string	Sí	Número telefónico mexicano de 10 dígitos (sin código de país)
amount	number	Sí	Monto de recarga en MXN (10-500)
reference	string	No	Tu ID de orden/referencia interno para seguimiento

Montos Disponibles (MXN):

10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 300, 500

Ejemplo de Solicitud:

```
curl -X POST https://latcom-fix-production.up.railway.app/api/enviadespens
-H "Content-Type: application/json" \
-H "x-api-key: haz_prod_2025" \
-H "x-customer-id: HAZ_001" \
-d '{
  "phone": "5566374683",
  "amount": 20,
  "reference": "ORDEN_12345"
}'
```

Respuesta Exitosa:

```
{
  "success": true,
  "transaction": {
    "id": "RLR1760353500123",
    "status": "SUCCESS",
    "amount_mxn": 20,
    "amount_usd": 1.08,
    "exchange_rate": 18.52,
    "phone": "5566374683",
    "reference": "ORDEN_12345",
    "operatorTransactionId": "LT62599623",
    "provider": "relier",
    "processedAt": "2025-10-13T10:00:00Z",
    "currency": "MXN"
  },
  "billing": {
    "deducted_usd": 1.08,
    "balance_before_usd": 10000.00,
    "balance_after_usd": 9998.92,
    "exchange_rate": 18.52
  },
  "message": "Recarga de 20 MXN procesada exitosamente. $1.08 USD deducido",
  "remaining_balance": 9998.92
}
```

4. Procesar Recarga (Asíncrona)

Envía una solicitud de recarga para procesamiento en segundo plano. Recomendado para operaciones de alto volumen.

Endpoint: `POST /api/enviadespensa/topup-async` **Autenticación:** Requerida

Límite de Tasa: 200 solicitudes por minuto

Cuerpo de la Solicitud: (Igual que la síncrona)

Respuesta:

```
{
  "success": true,
  "transaction": {
    "id": "RLR1760353500456",
    "status": "PENDING",
    "amount": 20,
    "phone": "5566374683",
    "reference": "ORDEN_12345",
    "queuedAt": "2025-10-13T10:00:00Z",
    "currency": "MXN"
  },
  "message": "Transacción en cola para procesamiento",
  "check_status_url": "/api/transaction/RLR1760353500456"
}
```

5. Consultar Estado de Transacción

Verifica el estado de una transacción específica.

Endpoint: `GET /api/transaction/{transactionId}` **Autenticación:** No requerida

(el ID de transacción actúa como token)

Ejemplo de Solicitud:

```
curl https://latcom-fix-production.up.railway.app/api/transaction/RLR17603
```

Respuesta:

```
{
  "success": true,
  "transaction": {
    "transaction_id": "RLR1760353500123",
    "customer_id": "HAZ_001",
    "phone": "5566374683",
    "amount_mxn": 20,
    "amount_usd": 1.08,
    "exchange_rate": 18.52,
    "status": "SUCCESS",
    "reference": "ORDEN_12345",
    "operator_transaction_id": "LT62599623",
    "provider": "relier",
    "response_time_ms": 2345,
    "created_at": "2025-10-13T10:00:00Z",
    "processed_at": "2025-10-13T10:00:02Z",
    "currency": "MXN"
  }
}
```

Estados de Transacción: - **PENDING** - Transacción enviada, esperando procesamiento - **SUCCESS** - Recarga completada exitosamente - **FAILED** - Recarga fallida (el saldo no será deducido)

Formato de Solicitud/Respuesta

Requisitos de Solicitud

1. **Content-Type:** Todas las solicitudes POST deben usar `Content-Type: application/json`
2. **Formato de Teléfono:** 10 dígitos, sin espacios ni caracteres especiales (ej., `5566374683`)
3. **Monto:** Debe ser uno de los montos MXN permitidos (10-500)
4. **Referencia:** Opcional pero recomendado para seguimiento

Formato de Respuesta

Todas las respuestas siguen esta estructura:

Éxito:

```
{
  "success": true,
  "transaction": { ... },
  "billing": { ... },
  "message": "Mensaje de éxito descriptivo"
}
```

Error:

```
{
  "success": false,
  "error": "Descripción del error",
  "details": { ... }
}
```

Manejo de Errores

Códigos de Estado HTTP Comunes

Código	Significado	Descripción
200	OK	Solicitud exitosa
400	Solicitud Incorrecta	Formato o parámetros de solicitud inválidos
401	No Autorizado	Credenciales de API inválidas o faltantes
403	Prohibido	Saldo insuficiente o IP no autorizada
429	Demasiadas Solicitudes	Límite de tasa excedido
500	Error Interno del Servidor	Error del sistema, contactar soporte
503	Servicio No Disponible	Sistema temporalmente no disponible

Ejemplos de Respuesta de Error

Credenciales Inválidas:

```
{
  "success": false,
  "error": "Credenciales de API inválidas"
}
```

Saldo Insuficiente:

```
{
  "success": false,
  "error": "Saldo insuficiente",
  "current_balance_usd": 5.00,
  "required_usd": 10.80
}
```

Monto Inválido:

```
{
  "success": false,
  "error": "Monto de producto inválido",
  "message": "Solo se permiten montos fijos",
  "allowed_amounts": [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 300],
  "requested_amount": 25
}
```

Límite de Tasa Excedido:

```
{
  "success": false,
  "error": "Demasiadas solicitudes de recarga. Máximo 200 por minuto."
}
```

Límite Diario Excedido:

```
{
  "success": false,
  "error": "Límite de transacciones diarias excedido",
  "daily_limit_mxn": 5000,
  "used_today_mxn": 4800,
  "requested_mxn": 300,
  "available_mxn": 200
}
```

Lógica de Reintento

Implemente retroceso exponencial para solicitudes fallidas:

1. **Errores de red:** Reintentar con retroceso (1s, 2s, 4s)
2. **429 Límite de Tasa:** Esperar 60 segundos antes de reintentar
3. **Errores 500/503:** Reintentar con retroceso (5s, 10s, 30s)
4. **Errores 400/401/403:** No reintentar, corregir la solicitud

Límites de Tasa y Seguridad

Límites de Tasa

Endpoint	Límite	Ventana
API General	300 solicitudes	15 minutos
Endpoints de recarga	200 solicitudes	1 minuto

Encabezados de Límite de Tasa:

```
X-RateLimit-Limit: 200
X-RateLimit-Remaining: 195
X-RateLimit-Reset: 1697193600
```

Límite de Transacciones Diarias

- **Máximo:** 5,000 MXN por período de 24 horas
- Previene gastos excesivos
- Se reinicia automáticamente a medianoche UTC

Características de Seguridad

1. **Autenticación con Clave API** - Todas las solicitudes validadas
2. **Solo HTTPS** - Todo el tráfico cifrado
3. **Limitación de Tasa** - Previene abusos
4. **Lista Blanca de IP** - Opcional (contactar soporte para habilitar)
5. **Registro de Transacciones** - Mantenimiento completo de auditoría

Mejores Prácticas

- Almacene credenciales en variables de entorno
 - Use HTTPS para todas las solicitudes
 - Implemente tiempos de espera de solicitud (30 segundos recomendado)
 - Registre todas las transacciones para reconciliación
 - Monitoree su saldo regularmente
 - Configure alertas de saldo bajo
-

Guía de Pruebas

Paso 1: Verificar Acceso a la API

```
curl https://latcom-fix-production.up.railway.app/health
```

Esperado: `{"status": "OK", ...}`

Paso 2: Consultar Su Saldo

```
curl -X GET https://latcom-fix-production.up.railway.app/api/balance \
-H "x-api-key: haz_prod_2025" \
-H "x-customer-id: HAZ_001"
```

Esperado: Saldo actual y límite de crédito

Paso 3: Probar Recarga Pequeña

```
curl -X POST https://latcom-fix-production.up.railway.app/api/enviadespens
-H "Content-Type: application/json" \
-H "x-api-key: haz_prod_2025" \
-H "x-customer-id: HAZ_001" \
-d '{
  "phone": "5566374683",
  "amount": 10,
  "reference": "PRUEBA_001"
}'
```

Esperado: Respuesta exitosa con detalles de la transacción

Paso 4: Verificar Transacción

```
curl https://latcom-fix-production.up.railway.app/api/transaction/RLR17603
```

Reemplace el ID de transacción con el del Paso 3.

Ejemplos de Código

Node.js / JavaScript

```
const axios = require('axios');

const RELIER_API_URL = 'https://latcom-fix-production.up.railway.app';
const API_KEY = 'haz_prod_2025';
const CUSTOMER_ID = 'HAZ_001';

async function procesarRecarga(telefono, monto, referencia) {
  try {
    const response = await axios.post(
      `${RELIER_API_URL}/api/enviadespensa/topup`,
      {
        phone: telefono,
        amount: monto,
        reference: referencia
      },
      {
        headers: {
          'Content-Type': 'application/json',
          'x-api-key': API_KEY,
          'x-customer-id': CUSTOMER_ID
        },
        timeout: 30000 // 30 segundos
      }
    );
  }
};
```

```

    if (response.data.success) {
      console.log('¡Recarga exitosa!');
      console.log('ID de Transacción:', response.data.transaction.id);
      console.log('Monto:', response.data.transaction.amount_mxn, 'MXN');
      console.log('Costo:', response.data.transaction.amount_usd, 'USD');
      console.log('Saldo restante:', response.data.remaining_balance, 'USD');
      return response.data;
    } else {
      console.error('Recarga fallida:', response.data.error);
      return null;
    }
  } catch (error) {
    if (error.response) {
      console.error('Error de API:', error.response.data.error);
      console.error('Estado:', error.response.status);
    } else {
      console.error('Error de Red:', error.message);
    }
    return null;
  }
}

// Consultar saldo
async function consultarSaldo() {
  try {
    const response = await axios.get(
      `${RELIER_API_URL}/api/balance`,
      {
        headers: {
          'x-api-key': API_KEY,
          'x-customer-id': CUSTOMER_ID
        }
      }
    );

    console.log('Saldo Actual:', response.data.current_balance, 'USD');
    return response.data;
  } catch (error) {

```



```
        console.error('Consulta de saldo fallida:', error.message);
        return null;
    }
}

// Uso
procesarRecarga('5566374683', 20, 'ORDEN_12345');
```

Python

```
import requests
import json

RELIER_API_URL = 'https://latcom-fix-production.up.railway.app'
API_KEY = 'haz_prod_2025'
CUSTOMER_ID = 'HAZ_001'

def procesar_recarga(telefono, monto, referencia):
    """Procesar una recarga móvil"""
    headers = {
        'Content-Type': 'application/json',
        'x-api-key': API_KEY,
        'x-customer-id': CUSTOMER_ID
    }

    payload = {
        'phone': telefono,
        'amount': monto,
        'reference': referencia
    }

    try:
        response = requests.post(
            f'{RELIER_API_URL}/api/enviadespensa/topup',
            headers=headers,
```

```

        json=payload,
        timeout=30
    )

    data = response.json()

    if data.get('success'):
        print(f"✅ ;Recarga exitosa!")
        print(f"ID de Transacción: {data['transaction']['id']}")
        print(f"Monto: {data['transaction']['amount_mxn']} MXN")
        print(f"Costo: ${data['transaction']['amount_usd']} USD")
        print(f"Saldo: ${data['remaining_balance']} USD")
        return data
    else:
        print(f"❌ Recarga fallida: {data.get('error')}")
        return None

except requests.exceptions.RequestException as e:
    print(f"❌ Error de solicitud: {str(e)}")
    return None

def consultar_saldo():
    """Consultar saldo de cuenta"""
    headers = {
        'x-api-key': API_KEY,
        'x-customer-id': CUSTOMER_ID
    }

    try:
        response = requests.get(
            f'{RELIER_API_URL}/api/balance',
            headers=headers,
            timeout=10
        )

        data = response.json()

        if data.get('success'):

```

```

        print(f"💰 Saldo: ${data['current_balance']} USD")
        return data
    else:
        print(f"❌ Consulta de saldo fallida: {data.get('error')}")
        return None

except requests.exceptions.RequestException as e:
    print(f"❌ Error de solicitud: {str(e)}")
    return None

# Uso
if __name__ == '__main__':
    consultar_saldo()
    procesar_recarga('5566374683', 20, 'ORDEN_12345')

```

PHP

```

<?php

define('RELIER_API_URL', 'https://latcom-fix-production.up.railway.app');
define('API_KEY', 'haz_prod_2025');
define('CUSTOMER_ID', 'HAZ_001');

function procesarRecarga($telefono, $monto, $referencia) {
    $url = RELIER_API_URL . '/api/enviadespensa/topup';

    $data = [
        'phone' => $telefono,
        'amount' => $monto,
        'reference' => $referencia
    ];

    $headers = [
        'Content-Type: application/json',
        'x-api-key: ' . API_KEY,
    ];
}

```

```

        'x-customer-id: ' . CUSTOMER_ID
    ];

    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_TIMEOUT, 30);

    $response = curl_exec($ch);
    $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    curl_close($ch);

    if ($httpCode === 200) {
        $result = json_decode($response, true);

        if ($result['success']) {
            echo "✅ ;Recarga exitosa!\n";
            echo "ID de Transacción: {$result['transaction']['id']}\n";
            echo "Monto: {$result['transaction']['amount_mxn']} MXN\n";
            echo "Costo: \${$result['transaction']['amount_usd']} USD\n";
            echo "Saldo: \${$result['remaining_balance']} USD\n";
            return $result;
        } else {
            echo "❌ Recarga fallida: {$result['error']}\n";
            return null;
        }
    } else {
        echo "❌ Error HTTP: $httpCode\n";
        return null;
    }
}

function consultarSaldo() {
    $url = RELIER_API_URL . '/api/balance';

    $headers = [

```

```

        'x-api-key: ' . API_KEY,
        'x-customer-id: ' . CUSTOMER_ID
    ];

    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_TIMEOUT, 10);

    $response = curl_exec($ch);
    $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    curl_close($ch);

    if ($httpCode === 200) {
        $result = json_decode($response, true);
        echo "💰 Saldo: \${$result['current_balance']} USD\n";
        return $result;
    } else {
        echo "❌ Consulta de saldo fallida\n";
        return null;
    }
}

// Uso
consultarSaldo();
procesarRecarga('5566374683', 20, 'ORDEN_12345');

?>

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

```

```

import java.time.Duration;
import org.json.JSONObject;

public class RelierAPI {
    private static final String RELIER_API_URL = "https://latcom-fix-produ
    private static final String API_KEY = "haz_prod_2025";
    private static final String CUSTOMER_ID = "HAZ_001";

    private static final HttpClient httpClient = HttpClient.newBuilder()
        .connectTimeout(Duration.ofSeconds(30))
        .build();

    public static JSONObject procesarRecarga(String telefono, int monto, S
    try {
        JSONObject payload = new JSONObject();
        payload.put("phone", telefono);
        payload.put("amount", monto);
        payload.put("reference", referencia);

        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create(RELIER_API_URL + "/api/enviadespensa/t
            .header("Content-Type", "application/json")
            .header("x-api-key", API_KEY)
            .header("x-customer-id", CUSTOMER_ID)
            .POST(HttpRequest.BodyPublishers.ofString(payload.toSt
            .timeout(Duration.ofSeconds(30))
            .build();

        HttpResponse<String> response = httpClient.send(request,
            HttpResponse.BodyHandlers.ofString());

        JSONObject result = new JSONObject(response.body());

        if (result.getBoolean("success")) {
            System.out.println("✅ ;Recarga exitosa!");
            JSONObject transaction = result.getJSONObject("transaction
            System.out.println("ID de Transacción: " + transaction.get
            System.out.println("Monto: " + transaction.getDouble("amou

```

```

        System.out.println("Costo: $" + transaction.getDouble("amo
        System.out.println("Saldo: $" + result.getDouble("remainin
        return result;
    } else {
        System.out.println("❌ Recarga fallida: " + result.getStri
        return null;
    }

} catch (Exception e) {
    System.out.println("❌ Error: " + e.getMessage());
    return null;
}

}

public static JSONObject consultarSaldo() {
    try {
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create(RELIER_API_URL + "/api/balance"))
            .header("x-api-key", API_KEY)
            .header("x-customer-id", CUSTOMER_ID)
            .GET()
            .timeout(Duration.ofSeconds(10))
            .build();

        HttpResponse<String> response = httpClient.send(request,
            HttpResponse.BodyHandlers.ofString());

        JSONObject result = new JSONObject(response.body());

        if (result.getBoolean("success")) {
            System.out.println("💰 Saldo: $" + result.getDouble("curre
            return result;
        } else {
            System.out.println("❌ Consulta de saldo fallida");
            return null;
        }

    } catch (Exception e) {

```

```
        System.out.println("❌ Error: " + e.getMessage());
        return null;
    }
}

public static void main(String[] args) {
    consultarSaldo();
    procesarRecarga("5566374683", 20, "ORDEN_12345");
}
}
```

Soporte

Soporte Técnico

Para asistencia técnica, soporte de integración o preguntas sobre la API:

- **Email:** support@relier.com
- **Tiempo de Respuesta:** Dentro de 24 horas
- **Soporte de Emergencia:** Disponible para problemas de producción

Gestión de Cuenta

Para facturación, recargas de saldo o consultas de cuenta:

- **Email:** accounts@relier.com
- **Alertas de Saldo:** Notificaciones automáticas en el umbral de \$1,000 USD

Estado del Sistema

Monitorear la salud del sistema y tiempo de actividad: - **Página de Estado:**

<https://latcom-fix-production.up.railway.app/health> - **SLA de Tiempo de Actividad:**

99.9%

Actualizaciones de Documentación

Este documento tiene control de versiones. Verifique las actualizaciones regularmente o contacte a soporte para la última versión.

Apéndice

Operadores Soportados

- Telcel
- Movistar / Telefonica
- AT&T México
- Unefon
- Virgin Mobile México

Denominaciones Disponibles (MXN)

Monto	Aprox. USD*
10	~\$0.54
20	~\$1.08
30	~\$1.62
40	~\$2.16
50	~\$2.70
60	~\$3.24
70	~\$3.78
80	~\$4.32
90	~\$4.86
100	~\$5.40
150	~\$8.10
200	~\$10.80
300	~\$16.20
500	~\$27.00

*Los costos en USD son aproximados y se basan en tipos de cambio en tiempo real

(≈18.5 MXN/USD)

Información de Tipo de Cambio

- Los tipos de cambio se actualizan en tiempo real
- Las tasas son competitivas y transparentes
- Detalles completos de forex incluidos en cada respuesta de transacción
- Tasas históricas disponibles a través de registros de transacciones

Reconciliación

- Registros de transacciones disponibles a través del panel de administración
- Funcionalidad de exportación CSV para contabilidad
- Informes diarios/mensuales disponibles bajo solicitud
- Todas las transacciones incluyen IDs únicos para seguimiento

Fin del Documento

Para preguntas sobre este documento o la API de Relier, por favor contacte al gerente de cuenta de HAZ Group o soporte técnico.

Última Actualización: 13 de Octubre, 2025 **Versión del Documento:** 1.0 **Versión de API:** v1