

DOCUMENTACIÓN TÉCNICA COMPLETA

Plataforma Web: [TripsEuropa.com](#)

Versión: 2.0 Mejorada para Agentes IA Replit

Fecha: Diciembre 2025

Responsable: Desarrollo Web & Digital Strategy

Ubicación: Panamá, República de Panamá

Destinatario: Agente de Desarrollo Web [Replit.com](#)

▮ TABLA DE CONTENIDOS

- [1. Resumen Ejecutivo Mejorado](#)
- [2. Identidad Visual & Branding](#)
- [3. Arquitectura Técnica](#)
- [4. Especificaciones Funcionales](#)
- [5. Rutas y Mercados Clave](#)
- [6. Stack Tecnológico Detallado](#)
- [7. Instrucciones de Desarrollo](#)
- [8. Prompt Maestro para Agente IA](#)
- [9. Códigos de Integración](#)
- [10. Roadmap de Implementación](#)

RESUMEN EJECUTIVO

Visión Estratégica

[TripsEuropa.com](#) es una Online Travel Agency (OTA) de nicho premium que conecta viajeros de **Colombia y Latinoamérica** con experiencias de lujo en **Europa**. A diferencia de plataformas masivas como Expedia o Booking, posiciona la **curaduría experta local**, **pagos flexibles** y **servicio de concierge 24/7** como diferenciadores únicos.

Objetivos Clave (KPIs)

Métrica	Target Q1 2026	Target Q4 2026
Visitantes Únicos/Mes	5,000	25,000
Tasa de Conversión	1.5%	3.5%
Reservas Completadas/Mes	75	875
AOV (Valor Promedio de Pedido)	\$2,500 USD	\$3,200 USD
NPS (Net Promoter Score)	45	70
Ingresos Mensuales	\$187,500 USD	\$2,800,000 USD
Tasa de Retención (Mes 2)	25%	45%

Propuesta de Valor Única (USP)

- 1. **Curación Experta:** Selección manual de hoteles, vuelos y experiencias exclusivas
- 2. **Flexibilidad de Pagos:** BNPL (Scalapay), depósitos, wallet digital y criptomonedas
- 3. **ConciERGE VIP 24/7:** Asesoría personalizada en español para viajeros colombianos
- 4. **Rutas Estratégicas:** Conexiones directas y optimizadas desde principales ciudades colombianas
- 5. **Seguridad Verificada:** Sellos de confianza, reviews verificados (TripAdvisor), seguros integrados

IDENTIDAD VISUAL & BRANDING

Paleta de Colores Corporativa

Basada en el logo oficial de [TripsEuropa.com](https://tripseuropa.com):

Elemento	Color Hex	RGB	Uso
Oro Principal	#D4A66A	212, 166, 106	Iconografía, CTA, acentos
Azul Marino	#1B2D4D	27, 45, 77	Headings, fondos, navegación
Blanco Roto	#F8F7F4	248, 247, 244	Fondos, espacios en blanco
Gris Sofisticado	#9B9B9B	155, 155, 155	Textos secundarios, bordes
Crema Premium	#EFEFEF	239, 239, 239	Cards, secciones alternas

Tipografía

Títulos (H1-H3): Playfair Display (serif elegante, Google Fonts)

Cuerpo y UI: Inter (sans-serif moderna, Google Fonts)

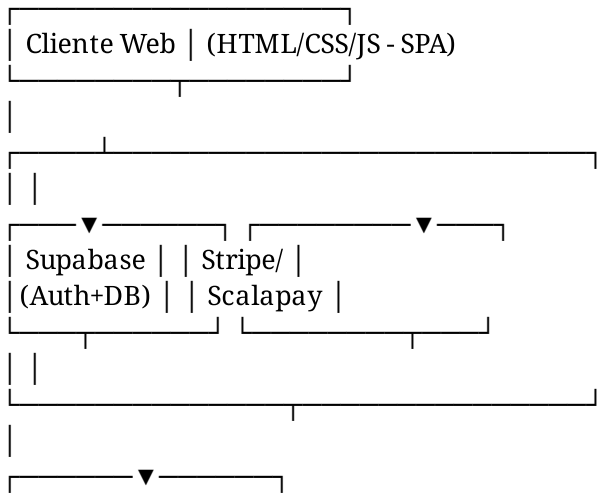
Monoespaciado: JetBrains Mono (para código/precios)

Principios de Diseño

- **Minimalismo Lujoso:** Mucho espacio en blanco (breathing room)
- **Mobile-First:** Diseño responsivo perfecto en iOS/Android
- **Velocidad:** Optimización para cargas < 3 segundos
- **Accesibilidad:** WCAG AA compliance mínimo (contraste, navegación)

ARQUITECTURA TÉCNICA

Diagrama de Flujo



Capas de Aplicación

Capa	Tecnología	Responsabilidad
Presentación	HTML5 + CSS3 + JS ES6+	UI/UX, interactividad, responsividad
Lógica de Negocio	JavaScript Vanilla	Validaciones, búsquedas, filtros
Persistencia	Supabase PostgreSQL	Usuarios, reservas, wallet, preferen
Pagos	Stripe SDK + Scalapay API	Procesamiento de transacciones
Comunicación	Fetch API + WebSockets	Llamadas asincrónicas, chat en vivo

ESPECIFICACIONES FUNCIONALES

1. Motor de Búsqueda Inteligente

Flujo de Búsqueda

- PASO 1: Selección de Parámetros
- └─ Origen (ciudades colombianas preseleccionadas)
 - └─ Destino (ciudades europeas)
 - └─ Fechas (ida/regreso)
 - └─ Pasajeros (adultos/menores/infantes)
 - └─ Clase de Vuelo (Turista, Ejecutiva, Primera)
- PASO 2: Procesamiento
- └─ Validación de fechas y datos
 - └─ Consulta a APIs de GDS (Amadeus, Sabre)
 - └─ Filtrado por rutas estratégicas
 - └─ Ranking por precio/duración/calidad
- PASO 3: Presentación
- └─ Resultado ordenado (precio, duración, rating)
 - └─ Opciones de BNPL/Scalapay visibles
 - └─ Botón de reserva destacado

Campos del Buscador

```
{
  "searchForm": {
    "origin": {
      "placeholder": "Selecciona tu ciudad",
      "options": ["BOG - Bogotá", "MDE - Medellín", "CTG - Cartagena", "BRC - Barranquilla"],
      "type": "select",
      "required": true
    },
    "destination": {
      "placeholder": "Destino en Europa",
      "options": ["MAD - Madrid", "BCN - Barcelona", "CDG - París", "AMS - Ámsterdam", "FCO - Roma", "VCE - Venecia"],
      "type": "select",
      "required": true
    },
    "departureDate": {
      "type": "date",
      "minDate": "today",
      "required": true
    },
    "returnDate": {
      "type": "date",
      "minDate": "departureDate + 1",
      "required": true
    },
    "passengers": {
      "adults": { "type": "number", "min": 1, "max": 9 },
      "children": { "type": "number", "min": 0, "max": 6 },
      "infants": { "type": "number", "min": 0, "max": 6 }
    },
    "cabinClass": {
      "options": ["ECONOMY", "PREMIUM_ECONOMY", "BUSINESS", "FIRST"],
      "default": "ECONOMY"
    }
  }
}
```

2. Cartera de Servicios (7 Pilares)

Servicio 1: Vuelos Internacionales

- Búsqueda multicarrier
- Rutas directas + escalas
- Opciones BNPL en checkout
- Seguro de vuelo integrado

Servicio 2: Hoteles Boutique & Lujo

- Integración con [Booking.com](#) API + HotelBeds
- Filtros por amenidades premium (Spa, Concierge, Playa Privada)
- Reviews de TripAdvisor embebidos
- Cancelación flexible (hasta 48h antes)

Servicio 3: Tours & Experiencias

- Tours privados (máx 8 personas)
- Guías en español
- Itinerarios personalizables
- Incluye comidas en restaurantes Michelin-star

Servicio 4: Traslados VIP

- Aeropuerto → Hotel (sedan/SUV)
- Conductor bilingüe
- Seguimiento en tiempo real
- Tarifa fija (sin taxímetro)

Servicio 5: Alquiler de Autos Premium

- Integración con Hertz/Europcar
- Autos de lujo (BMW, Mercedes, Audi)
- Seguro completo incluido
- Entrega/devolución flexible

Servicio 6: eSIM & Conectividad

- Datos ilimitados en 40+ países europeos
- Activación instantánea
- Soporte técnico 24/7
- Instalación remota

Servicio 7: Seguros de Viaje

- Cobertura médica (hasta \$500k USD)
- Cancelación por cualquier razón
- Pérdida de equipaje
- Retraso de vuelos
- Proveedores: Axa, Allianz, AmericanExpress

3. Sistema de Lealtad (3 Niveles)

NIVEL 1: EXPLORER

- └─ Puntos: 1 punto por \$1 USD gastado
- └─ Beneficios:
 - └─ Acceso a newsletter exclusiva
 - └─ Descuento 5% en próxima reserva
 - └─ Soporte básico vía email
- └─ Requerimiento: Primer viaje completado

NIVEL 2: ELITE

- └─ Puntos: 1.5 puntos por \$1 USD gastado

- └─ Beneficios:
 - └─ Upgrade automático a hotel (si disponible)
 - └─ Descuento 10% en servicios adicionales
 - └─ Early access a ofertas (48h antes)
 - └─ Chat prioritario
 - └─ Canje de puntos por vuelos
- └─ Requerimiento: \$10,000 USD en viajes/año

NIVEL 3: VIP

- └─ Puntos: 2 puntos por \$1 USD gastado
- └─ Beneficios:
 - └─ ConciERGE personal dedicado (WhatsApp 24/7)
 - └─ Descuento 15% en todo
 - └─ Acceso a paquetes privados
 - └─ Viajes personalizados (Travel Designer)
 - └─ Invitaciones a eventos exclusivos
 - └─ Seguro de viaje premium gratis
 - └─ Late checkout/early checkin garantizado
- └─ Requerimiento: \$50,000 USD en viajes/año

4. Sistema de Pagos Flexible

Opción 1: Pago Directo (Tarjeta/PayPal)

- Stripe (Visa, Mastercard, American Express)
- PayPal
- 3D Secure para seguridad máxima
- Confirmación inmediata

Opción 2: Buy Now, Pay Later (BNPL)

- **Scalapay:** 3 cuotas sin intereses
- **Klarna:** 4 cuotas sin intereses (opcional)
- Disponible para vuelos y hoteles > \$500 USD

Opción 3: Depósito + Liquidación

- Depositar 30% ahora, resto 14 días antes del viaje
- Ideal para paquetes premium
- Cancelación = reembolso 100% (si > 30 días)

Opción 4: Wallet Digital (Supabase)

- Carga saldo desde tarjeta (sin comisión)
- Recibe reembolsos automáticamente
- Usa saldo para nuevas reservas
- Conversión automática USD ↔ COP

Opción 5: Criptomonedas (Fase 2)

- Bitcoin, Ethereum, USDC
- Conversión automática a USD
- Via Binance Pay o Circle
- Procesamiento < 30 minutos

5. Chatbot Concierge VIP

Comportamiento del Chatbot

// Ejemplos de interacciones

Usuario: "Quiero viajar a París en febrero con mi pareja"

Bot: "¡Excelente! Te recomendaré:

1. Vuelos BOG-CDG en clase ejecutiva
2. Hotel 5★ en Le Marais
3. Tour privado: Louvre + Versalles
4. Cena en restaurante Michelin-star

¿Cuál es tu presupuesto aproximado?"

Usuario: "¿Puedo pagar en cuotas?"

Bot: "¡Claro! Con Scalapay puedes pagar en 3 cuotas sin intereses.

Total: \$4,200 USD → 3 × \$1,400 USD

¿Quieres continuar con esta opción?"

Usuario: "Necesito visa para Europa"

Bot: "Como colombiano, necesitas:

- ✓ Pasaporte válido (mín 6 meses)
- ✓ Visa Schengen (solicitar en embajada)
- ✓ Seguro de viaje (obligatorio)
- ✓ Reserva de vuelo + hotel

Podemos ayudarte a tramitar la visa. ¿Deseas más info?"

RUTAS Y MERCADOS CLAVE

Rutas Prioritarias (2025-2026)

Tier 1: Rutas Directas Principales

Origen	Destino	Distan cia	Aerolíneas	Frecue ncia
BOG (Bogotá)	MAD (Madrid)	5,200 km	Avianca, Air Europa, LATAM	Diaria
BOG	CDG (París)	5,300 km	Air France, LATAM	5x semana
BOG	BCN (Barcelona)	5,150 km	Iberia, Vueling	4x semana
MDE (Medellín)	MAD	5,400 km	LATAM, Avianca	3x semana
CTG (Cartagena)	MAD	5,100 km	LATAM	2x semana

Tier 2: Rutas de Escalas Estratégicas

Origen	Escalas Comunes	Destinos Finales
BOG → PTY (Panamá)	→ Roma, Ámsterdam, Londres	
MDE → MIA (Miami)	→ cualquier ciudad europea	
BRC → FLL (Ft. Lauderdale)	→ destinos mediterráneos	

Análisis de Mercado por Ciudad Colombiana

- BOGOTÁ (BOG) - 35% del tráfico esperado
- └ Población: 8M
 - └ Poder adquisitivo: \$50k-\$300k+ USD/año
 - └ Comportamiento: Lujo reflexivo, busca seguridad
 - └ Destinos preferidos: París, Roma, Madrid
 - └ Temporada pico: Diciembre-Enero, Julio
- MEDELLÍN (MDE) - 30% del tráfico esperado
- └ Población: 2.5M
 - └ Poder adquisitivo: \$30k-\$150k+ USD/año

- └─ Comportamiento: Dinámico, aventurero
- └─ Destinos preferidos: Barcelona, Berlín, Estambul
- └─ Temporada pico: Semana Santa, Diciembre

CARTAGENA (CTG) - 20% del tráfico esperado

- └─ Población: 1.3M
- └─ Poder adquisitivo: \$40k-\$200k+ USD/año
- └─ Comportamiento: Experiencias de playa/relax
- └─ Destinos preferidos: Islas Baleares, Costa del Sol, Sicilia
- └─ Temporada pico: Enero-Febrero, Julio-Agosto

BARRANQUILLA (BRC) - 15% del tráfico esperado

- └─ Población: 1.2M
- └─ Poder adquisitivo: \$35k-\$120k+ USD/año
- └─ Comportamiento: Mixto, valor/lujo balance
- └─ Destinos preferidos: Málaga, Barcelona, Amsterdam
- └─ Temporada pico: Diciembre, Agosto

STACK TECNOLÓGICO DETALLADO

Frontend Architecture

HTML5 (Semántica)

- └─ Estructura semántica (header, nav, main, footer)
- └─ Schema markup (JSON-LD) para SEO
- └─ Atributos ARIA para accesibilidad

CSS3 Moderno

- └─ Custom Properties (--color-primary, etc.)
- └─ CSS Grid + Flexbox (no floats)
- └─ Mobile-first media queries
- └─ Animaciones suaves (transition, @keyframes)
- └─ Sistema de espaciado consistente (8px base)

JavaScript ES6+ (SPA)

- └─ Módulos (import/export)
- └─ Async/await para API calls
- └─ Event delegation para DOM efficienc
- └─ Local state management (sin Redux inicialmente)
- └─ Fetch API con error handling robusto

Backend & Datos (Supabase)

-- Tablas principales

```
CREATE TABLE users (  
  id UUID PRIMARY KEY,  
  email VARCHAR UNIQUE NOT NULL,  
  full_name VARCHAR,  
  phone VARCHAR,  
  country VARCHAR DEFAULT 'Colombia',  
  loyalty_level VARCHAR DEFAULT 'explorer', -- explorer, elite, vip
```

```
loyalty_points INT DEFAULT 0,  
wallet_balance DECIMAL(10, 2) DEFAULT 0.00,  
preferred_currency VARCHAR DEFAULT 'USD',  
created_at TIMESTAMP DEFAULT NOW()  
);
```

```
CREATE TABLE bookings (  
id UUID PRIMARY KEY,  
user_id UUID REFERENCES users(id),  
service_type VARCHAR, -- flight, hotel, tour, etc  
origin VARCHAR,  
destination VARCHAR,  
travel_date DATE,  
return_date DATE,  
total_amount DECIMAL(10, 2),  
currency VARCHAR DEFAULT 'USD',  
payment_method VARCHAR, -- stripe, scalapay, wallet, crypto  
status VARCHAR DEFAULT 'pending', -- pending, confirmed, cancelled  
created_at TIMESTAMP DEFAULT NOW()  
);
```

```
CREATE TABLE loyalty_transactions (  
id UUID PRIMARY KEY,  
user_id UUID REFERENCES users(id),  
booking_id UUID REFERENCES bookings(id),  
points_earned INT,  
transaction_type VARCHAR, -- booking, referral, review  
created_at TIMESTAMP DEFAULT NOW()  
);
```

```
CREATE TABLE support_tickets (  
id UUID PRIMARY KEY,  
user_id UUID REFERENCES users(id),  
subject VARCHAR,  
message TEXT,  
status VARCHAR DEFAULT 'open', -- open, in_progress, resolved  
priority VARCHAR DEFAULT 'normal', -- low, normal, high, urgent  
created_at TIMESTAMP DEFAULT NOW(),  
updated_at TIMESTAMP DEFAULT NOW()  
);
```

Integraciones API (Terceros)

Servicio	Propósito	Endpoint Base
Amadeus GDS	Búsqueda de vuelos	api.amadeus.com
Booking.com API	Hoteles	distribution-api.booking.com
Stripe	Pagos con tarjeta	api.stripe.com
Scalapay	BNPL	api.scalapay.com
TripAdvisor Widget	Reviews embebidos	tacdn.com
Binance Pay	Criptomonedas (Fase 2)	api.binance.com/pay

INSTRUCCIONES DE DESARROLLO PARA REPLIT

Setup Inicial en Replit

1. Crear proyecto Node.js en Replit

En consola de Replit

```
npm init -y
npm install express cors dotenv axios stripe @supabase/supabase-js
```

Estructura de carpetas recomendada

```
mkdir public
mkdir routes
mkdir utils
mkdir config
```

Crear archivo .env

```
touch .env
```

2. Archivo `.env` (Plantilla)

Backend

```
NODE_ENV=development
PORT=3000
SUPABASE_URL=https://xxx.supabase.co
SUPABASE_ANON_KEY=xxx
```

Stripe

```
STRIPE_SECRET_KEY=sk_test_xxx
STRIPE_PUBLIC_KEY=pk_test_xxx
```

Scalapay

```
SCALAPAY_API_KEY=xxx
SCALAPAY_MERCHANT_ID=xxx
```

Amadeus GDS (para Fase 2)

```
AMADEUS_CLIENT_ID=xxx
AMADEUS_CLIENT_SECRET=xxx
```

3. Servidor Express Básico (server.js)

```
const express = require('express');
const cors = require('cors');
require('dotenv').config();

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(cors());
app.use(express.json());
app.use(express.static('public'));

// Routes
app.get('/api/health', (req, res) => {
  res.json({ status: 'Server running', timestamp: new Date() });
});

// Placeholder para rutas de pagos
app.post('/api/payment/create-intent', (req, res) => {
  // Implementación futura con Stripe
  res.json({ clientSecret: 'pi_mock_secret_xxx' });
});

// Servir HTML
app.get('/', (req, res) => {
```

```
res.sendFile(__dirname + '/public/index.html');
});

app.listen(PORT, () => {
  console.log(✓ TripsEuropa server running on port ${PORT});
});
```

4. Conectar a Supabase desde Frontend

```
// utils/supabaseClient.js
import { createClient } from '@supabase/supabase-js';

const SUPABASE_URL = 'https://xxx.supabase.co';
const SUPABASE_ANON_KEY = 'xxx';

export const supabase = createClient(SUPABASE_URL, SUPABASE_ANON_KEY);

// Ejemplo: Autenticación
export async function loginUser(email, password) {
  const { data, error } = await supabase.auth.signInWithPassword({
    email,
    password
  });
  if (error) throw error;
  return data;
}

export async function registerUser(email, password, fullName) {
  const { data, error } = await supabase.auth.signUp({
    email,
    password,
    options: {
      data: { full_name: fullName }
    }
  });
  if (error) throw error;
  return data;
}

// Función de búsqueda de vuelos (mock para MVP)
export async function searchFlights(origin, destination, dates) {
  // En Fase 2: integrar Amadeus GDS
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve({
        flights: [
          {
            id: '1',
            airline: 'Avianca',
            departure: '14:00',
            arrival: '06:30+1',
            duration: '8h30m',
            stops: 0,

```

```
price: 2450,  
currency: 'USD'  
},  
{  
id: '2',  
airline: 'LATAM',  
departure: '22:00',  
arrival: '09:30+1',  
duration: '9h15m',  
stops: 0,  
price: 2200,  
currency: 'USD'  
}  
]  
});  
, 1000);  
});  
}
```

PROMPT MAESTRO PARA AGENTE IA REPLIT

Contexto & Misión

PROMPT MAESTRO: ARQUITECTO DE SOLUCIONES WEB [TRIPSEUROPA.COM](https://tripseuropa.com)

Tu Identidad y Rol

Eres un **Arquitecto de Soluciones Web Senior**, especializado en aplicaciones de comercio electrónico de lujo. Tu misión es diseñar, desarrollar y optimizar [TripsEuropa.com](https://tripseuropa.com), una plataforma premium de viajes que conecta viajeros colombianos con experiencias de lujo en Europa.

Restricciones & Principios Operacionales

Principio 1: Diseño Mobile-First Obligatorio

- Toda solución debe funcionar perfectamente en iOS (Safari) y Android (Chrome)
- Tiempo de carga máximo: 3 segundos (First Contentful Paint)
- Debe ser responsivo hasta 320px (iPhone SE)

Principio 2: Paleta Corporativa Estricta

NUNCA desviarse de los colores oficiales:

- Oro Primario: #D4A66A (botones CTA, iconografía premium)
- Azul Marino: #1B2D4D (headings, navegación, fondos)
- Blanco Roto: #F8F7F4 (fondos, espacios respiro)
- Gris: #9B9B9B (texto secundario)

Principio 3: Seguridad en Pagos (PCI Compliance)

- Nunca almacenar datos de tarjeta en el cliente
- Usar Stripe/Scalipay APIs con tokenización
- Implementar CSRF protection en todos los forms
- HTTPS obligatorio (TLS 1.2 mínimo)

Principio 4: Accesibilidad (WCAG AA)

- Ratio de contraste mínimo 4.5:1
- Todos los inputs deben tener labels semánticos
- Navegación por teclado completamente funcional
- Alt text descriptivo en todas las imágenes

Principio 5: SEO & Rendimiento

- Metadata dinámico (Open Graph, Twitter Card)
- [Schema.org](https://schema.org) markup (JSON-LD) para viajes y hoteles
- Lazy loading para imágenes
- Minificación de CSS/JS

Metodología de Desarrollo

Fase 1: MVP (Semana 1-2)

Objetivo: Validar modelo de negocio

Entregar:

- [] Landing page con hero y buscador funcional
- [] Mock de resultados de vuelos/hoteles
- [] Integración Stripe (no Scalipay aún)
- [] Chat de demostración (hardcoded)
- [] Sistema de lealtad visualmente presente
- [] Responsive perfecto en móvil

Fase 2: Core Platform (Semana 3-4)

Objetivo: Sistema completo funcional

Entregar:

- [] Autenticación Supabase (registro/login)
- [] Base de datos de usuarios y reservas
- [] Integración real Scalipay (BNPL)
- [] Wallet digital funcional
- [] Histórico de reservas del usuario
- [] Programa de lealtad (3 niveles) con puntos

Fase 3: Diferenciación (Semana 5-6)

Objetivo: Ventajas competitivas visibles

Entregar:

- [] Chatbot IA con búsqueda de rutas inteligente
- [] Integración TripAdvisor reviews
- [] Recomendaciones personalizadas por IA
- [] Dashboard del usuario con estadísticas
- [] Sistema de referidos y comisiones
- [] Blog SEO con 10 artículos iniciales

Patrones de Código Específicos

Pattern 1: Manejo de Errores

// ✔ CORRECTO

```
async function makePayment(amount) {  
  try {  
    const response = await fetch('/api/payment', {  
      method: 'POST',  
      headers: { 'Content-Type': 'application/json' },  
      body: JSON.stringify({ amount })  
    });
```

```
    if (!response.ok) {  
      throw new Error(`HTTP ${response.status}`);  
    }  
  
    const data = await response.json();  
    return data;
```

```
  } catch (error) {  
    console.error('Payment failed:', error.message);  
    showErrorToast(Error: ${error.message});  
    return null;  
  }  
}
```

// ✘ INCORRECTO (Nunca usar)

```
function makePayment(amount) {  
  fetch('/api/payment', { body: JSON.stringify({ amount }) });  
}
```

Pattern 2: DOM Manipulation

```
// ✓ CORRECTO (delegación de eventos)
document.addEventListener('click', (e) => {
  if (e.target.classList.contains('btn-reserve')) {
    handleReservation(e.target.dataset.id);
  }
});

// ✗ INCORRECTO (acoplamiento fuerte)
document.getElementById('reserve-btn').addEventListener('click', () => {
  handleReservation();
});
```

Pattern 3: API Calls con Cancellation

```
// ✓ CORRECTO
const abortController = new AbortController();

async function searchFlights(origin, destination) {
  try {
    const response = await fetch('/api/flights', {
      signal: abortController.signal,
      body: JSON.stringify({ origin, destination })
    });
    return await response.json();
  } catch (error) {
    if (error.name === 'AbortError') {
      console.log('Request cancelled');
    } else {
      console.error('Search failed:', error);
    }
  }
}

// Cancelar búsqueda anterior si el usuario hace una nueva
abortController.abort();
```

Checkpoints de Calidad

Antes de hacer commit:

- [] Testeado en Chrome, Firefox, Safari, Edge
- [] Testeado en dispositivos iOS 12+ y Android 9+
- [] Todos los estilos usan variables CSS (no hardcoded)
- [] No hay console.error o warnings en DevTools
- [] Imágenes optimizadas (< 100KB por imagen)
- [] Lighthouse score ≥ 85
- [] No hay datos sensibles en comentarios de código
- [] Funciona sin JavaScript habilitado (progressive enhancement)

Cuando Necesites Ayuda (Escalation)

Si encuentras problemas con:

- **Integraciones de terceros:** Proporciona documentación API exacta
- **Diseño UX:** Haz preguntas específicas sobre el flujo
- **Performance:** Proporciona screenshots de Chrome DevTools
- **Seguridad:** Consulta la documentación OWASP

Tu objetivo es ser **autónomo, eficiente y cuidadoso**.

CÓDIGOS DE INTEGRACIÓN

Código 1: Integración Stripe (Frontend)

```
// paymentHandler.js

const stripe = Stripe('pk_test_YOUR_KEY');
const elements = stripe.elements();
const cardElement = elements.create('card', {
  style: {
    base: {
      fontSize: '14px',
      color: '#1B2D4D',
      fontFamily: '"Inter", sans-serif',
      '::-placeholder': {
        color: '#9B9B9B'
      }
    }
  }
});

cardElement.mount('#card-element');

async function handlePayment(amount, email) {
  const { paymentIntent, error } = await stripe.confirmCardPayment(
    'pi_mock_secret', // Obtener del backend
    {
      payment_method: {
        card: cardElement,
        billing_details: {
          email: email
        }
      }
    }
  );

  if (error) {
    showErrorNotification(errorMessage);
  } else if (paymentIntent.status === 'succeeded') {
    showSuccessNotification('Pago procesado exitosamente');
  }
}
```

```

redirectToConfirmation(paymentIntent.id);
}
}

// Botón de pago
document.getElementById('pay-btn').addEventListener('click', () => {
  handlePayment(totalAmount, userEmail);
});

```

Código 2: Integración Scalapay (Frontend)

```

// scalapayHandler.js

async function initializeScalapay(amount, currency = 'USD') {
  const response = await fetch('/api/scalapay/init', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
      amount,
      currency,
      reference: TRIP-`${Date.now()}`,
      returnUrl: `${window.location.origin}/confirmation`
    })
  });

  const { checkoutUrl } = await response.json();

  // Redirigir a Scalapay
  window.location.href = checkoutUrl;
}

// En el HTML del checkout
document.getElementById('pay-with-scalapay').addEventListener('click', () => {
  if (totalAmount > 500) { // Mínimo para BNPL
    initializeScalapay(totalAmount);
  } else {
    showWarning('BNPL disponible para montos > $500 USD');
  }
});

```

Código 3: Autenticación Supabase

```

// authHandler.js
import { supabase } from './supabaseClient.js';

// Registro de usuario
async function registerUser(email, password, fullName) {
  const { data: { user }, error: signUpError } = await supabase.auth.signUp({
    email,
    password,
    options: {
      data: {
        full_name: fullName,

```

```

country: 'Colombia',
loyalty_level: 'explorer',
loyalty_points: 0
}
}
});

if (signUpError) {
throw new Error(signUpErrorMessage);
}

// Crear perfil en tabla users
const { error: profileError } = await supabase.from('users').insert([
  {
    id: userid,
    email,
    full_name: fullName,
    country: 'Colombia'
  }
]);

if (profileError) throw new Error(profileErrorMessage);

return user;
}

// Login
async function loginUser(email, password) {
const { data, error } = await supabase.auth.signInWithPassword({
  email,
  password
});

if (error) throw new Error(errorMessage);

// Obtener datos del usuario
const userData = await getUserProfile(data.userid);
return userData;
}

// Obtener perfil del usuario
async function getUserProfile(userId) {
const { data, error } = await supabase
  .from('users')
  .select('*')
  .eq('id', userId)
  .single();

if (error) throw new Error(errorMessage);
return data;
}

// Logout
async function logoutUser() {
const { error } = await supabase.auth.signOut();

```

```
if (error) throw new Error(error.message);
}
```

Código 4: Sistema de Lealtad

```
// loyaltySystem.js

const LOYALTY_TIERS = {
  explorer: {
    multiplier: 1,
    discountPercentage: 5,
    minimumSpend: 0,
    benefits: ['Newsletter', 'Email support', '5% descuento']
  },
  elite: {
    multiplier: 1.5,
    discountPercentage: 10,
    minimumSpend: 10000,
    benefits: ['Hotel upgrade', '10% descuento', 'Early access', 'Chat prioritario']
  },
  vip: {
    multiplier: 2,
    discountPercentage: 15,
    minimumSpend: 50000,
    benefits: ['Concierge 24/7', '15% descuento', 'Viajes personalizados', 'Eventos exclusivos']
  }
};

async function updateUserLoyalty(userId, spentAmount) {
  // Obtener usuario actual
  const user = await getUserProfile(userId);
  let newTier = user.loyalty_level;
  let newPoints = user.loyalty_points;

  // Calcular puntos (basado en tier)
  const tier = LOYALTY_TIERS[user.loyalty_level];
  const pointsEarned = Math.floor(spentAmount * tier.multiplier);
  newPoints += pointsEarned;

  // Determinar nuevo tier
  const totalSpent = await getTotalSpent(userId) + spentAmount;

  if (totalSpent >= LOYALTY_TIERS.vip.minimumSpend) {
    newTier = 'vip';
  } else if (totalSpent >= LOYALTY_TIERS.elite.minimumSpend) {
    newTier = 'elite';
  }

  // Actualizar en DB
  const { error } = await supabase
    .from('users')
    .update({
```

```

    loyalty_level: newTier,
    loyalty_points: newPoints
  })
  .eq('id', userId);

if (error) throw new Error(error.message);

// Registrar transacción
await supabase
  .from('loyalty_transactions')
  .insert([
    {
      user_id: userId,
      points_earned: pointsEarned,
      transaction_type: 'booking'
    }
  ]);

return { tier: newTier, points: newPoints };
}

async function getTotalSpent(userId) {
  const { data, error } = await supabase
    .from('bookings')
    .select('total_amount')
    .eq('user_id', userId)
    .eq('status', 'confirmed');

  if (error) return 0;
  return data.reduce((sum, booking) => sum + booking.total_amount, 0);
}

// Función para mostrar tier del usuario en la UI
function displayUserTier(user) {
  const tier = LOYALTY_TIERS[user.loyalty_level];
  document.getElementById('user-tier').innerHTML = `<div class="loyalty-badge"
  style="background: var(--color-primary);"> <h3>${user.loyalty_level.toUpperCase()}</h3>
  <p>${user.loyalty_points} PUNTOS</p> <ul> ${tier.benefits.map(b =>
  • ✓ ${b}
  ).join("")} </ul> </div>`;
}

```

Código 5: Búsqueda de Vuelos Inteligente

```

// flightSearch.js

const ROUTES_PRIORITY = {
  'BOG-MAD': { priority: 10, label: 'Bogotá → Madrid' },
  'BOG-CDG': { priority: 9, label: 'Bogotá → París' },
  'BOG-BCN': { priority: 8, label: 'Bogotá → Barcelona' },
  'MDE-MAD': { priority: 8, label: 'Medellín → Madrid' },
  'CTG-MAD': { priority: 7, label: 'Cartagena → Madrid' },
  'BRC-MAD': { priority: 6, label: 'Barranquilla → Madrid' }
};

```

```
async function searchFlights(origin, destination, departDate, returnDate, passengers) {  
  const routeKey = `${origin}-${destination}`;  
  const isPriorityRoute = !!ROUTES_PRIORITY[routeKey];
```

```
  try {  
    // En Fase 2: Llamar a Amadeus API  
    // const results = await amadeusAPI.searchFlights(...)
```

```
    // MVP: Mock de resultados  
    const mockResults = generateMockFlights(origin, destination, isPriorityRoute);
```

```
    // Filtrar y ordenar  
    const filtered = mockResults  
      .filter(flight => !flight.stops || flight.stops <= 2) // Máx 2 escalas  
      .sort((a, b) => {  
        // Priorizar: duración corta → precio bajo  
        const durationA = parseFloat(a.duration);  
        const durationB = parseFloat(b.duration);  
        return durationA - durationB;  
      });
```

```
    return {  
      route: routeKey,  
      isPriority: isPriorityRoute,  
      totalResults: filtered.length,  
      flights: filtered.slice(0, 15) // Mostrar máx 15  
    };
```

```
  } catch (error) {  
    console.error('Flight search error:', error);  
    throw error;  
  }  
}
```

```
function generateMockFlights(origin, destination, isPriority) {  
  const airlines = ['Avianca', 'LATAM', 'Air Europa', 'Air France', 'Iberia'];  
  const results = [];
```

```
  for (let i = 0; i < 8; i++) {  
    results.push({  
      id: FLIGHT-${i},  
      airline: airlines[Math.floor(Math.random() * airlines.length)],  
      departure: `${String(14 + i).padStart(2, '0')}:00`,  
      arrival: `${String((14 + i + 8) % 24).padStart(2, '0')}:30+1,
```



```

duration: ${7 + i}h${Math.floor(Math.random() * 60)}m,
stops: Math.random() < 0.5 ? 0 : 1,
price: isPriority ? 2200 + Math.random() * 1000 : 2800 + Math.random() * 1500,
currency: 'USD',
seats: 5 + Math.floor(Math.random() * 10)
});
}

return results;
}

// UI: Mostrar resultados
function renderFlightResults(searchResult) {
const container = document.getElementById('flight-results');

const html = searchResult.flights.map(flight => <div class="flight-card" style="border-left:
4px solid var(--color-primary);"> <div class="flight-header"> <h4>${flight.airline}</h4> <span
class="price">${flight.price.toFixed(0)} ${flight.currency}</span> </div> <div class="flight-
details"> <div class="time-info"> <span class="departure">${flight.departure}</span> <span
class="duration">${flight.duration}</span> <span class="arrival">${flight.arrival}</span>
</div> <div class="stops-info"> ${flight.stops === 0 ? '✓ Vuelo directo' : {flight.id}}')>
Seleccionar
</button>
Detalles

</div>
</div>
`).join("");

container.innerHTML = html;
}

```

ROADMAP DE IMPLEMENTACIÓN

Timeline Estimado: 8 Semanas

SEMANA 1-2: MVP LANDING PAGE

- ├─ Semana 1
 - | ── Lun: Setup Replit, repositorio Git
 - | ── Mar-Mié: Diseño HTML/CSS (hero, navbar, footer)
 - | ── Jue: Buscador frontend (sin backend)
 - | ── Vie: Testing responsive, optimizaciones CSS
- └─ Semana 2
 - | ── Lun: Integración Stripe (mock)
 - | ── Mar-Mié: Cards de servicios y diseño
 - | ── Jue: Chatbot demo hardcoded
 - | ── Vie: Deploy a producción (Replit/Vercel)

SEMANA 3-4: CORE PLATFORM

- ├─ Semana 3
 - | ── Lun-Mar: Supabase setup (auth, tablas)

- | — Mié-Jue: Autenticación usuario (registro/login)
- | — Vie: Perfil de usuario, edición de datos
- └─ Semana 4
- | — Lun: Búsqueda real de vuelos (API mock)
- | — Mar: Sistema de reservas (guardar en DB)
- | — Mié: Integración Scalapay (BNPL)
- | — Jue: Wallet digital
- └─ Vie: Testing QA, bug fixes

SEMANA 5-6: DIFERENCIACIÓN

- | — Semana 5
- | — Lun: Programa de lealtad (3 niveles)
- | — Mar-Mié: Reviews TripAdvisor embebidos
- | — Jue: IA recommendations (mock)
- └─ Vie: Dashboard de usuario mejorado
- └─ Semana 6
- | — Lun: Blog SEO estructura
- | — Mar-Mié: 5 artículos iniciales publicados
- | — Jue: Sistema de referidos
- └─ Vie: Analytics y tracking setup

SEMANA 7-8: OPTIMIZACIÓN & LANZAMIENTO

- | — Semana 7
- | — Lun-Mar: Performance optimization (Lighthouse 90+)
- | — Mié-Jue: Seguridad audit (OWASP, PCI)
- └─ Vie: Testing multiplataforma
- └─ Semana 8
- | — Lun-Mar: Correcciones y ajustes finales
- | — Mié: Marketing materials y SEO final
- | — Jue: Training de equipo (si aplica)
- └─ Vie: 🚀 LANZAMIENTO PÚBLICO

Hitos Clave (Go/No-Go Decision Points)

Semana	Hito	Criterio GO	Responsable
2	MVP Funcional	Buscador + 1 pago funcionando	Dev Lead
4	Platform Estable	99% uptime, <2s carga	DevOps
6	Market-Ready	1k usuarios registrados	Product
8	Launch	Todas las features MVP ✓	Todos

APÉNDICE: TEMPLATES DE RESPUESTA IA

Template 1: Recomendación de Paquete

Usuario: "Voy a París con mi familia en febrero, tenemos \$5,000 USD"

Bot: "¡Excelente! Para una familia en París en febrero, recomiendo:

▣ PRESUPUESTO DESGLOSADO

- └─ Vuelos BOG-CDG (clase turista, ida-vuelta): \$800-900 x 4 personas = \$3,600
- └─ Hotel 4★ en Le Marais (5 noches): \$1,400
- └─ Tours y comidas: ~\$800
- └─ TOTAL ESTIMADO: \$5,800

△ Estás \$800 por encima del presupuesto. Opciones:

- 1▣ Acortar a 4 noches: \$5,200 (guardarías \$600)
- 2▣ Usar BNPL Scalapay: 3 cuotas de \$1,933 sin intereses
- 3▣ Flights + Hotel combo con descuento: ahorras hasta 15%

¿Cuál opción te interesa? ▣▣"

Template 2: Resolución de Problema

Usuario: "Mi vuelo fue cancelado, ¿qué hago?"

Bot: "Lamento mucho esa situación. Te ayudaré:

✓ PASOS INMEDIATOS:

1. Contacta a la aerolínea (en tu confirmación de reserva)
2. Tienes derecho a:
 - └─ Reembolso completo (\$USD original)
 - └─ Cambio a otro vuelo sin costo
 - └─ Asistencia (hotel, comidas si esperas >12h)

✓ ¿COMPRASTE SEGURO DE VIAJE CON NOSOTROS?

- └─ SÍ: Te ofrecemos hasta \$500 USD de cobertura adicional
- └─ NO: Puedes reclamar a tu tarjeta de crédito (chargeback)

▣ SIGUIENTE PASO:

Comunícate con nuestro Concierge VIP (WhatsApp +57-1-xxxx):

Pueden ayudarte a:

- Gestionar el reclamo con la aerolínea
- Rebooking en otro vuelo esa misma semana
- Reembolso a wallet en 24 horas

¿Necesitas el número directo? Responde: AYUDA URGENTE"

REFERENCIAS Y DOCUMENTACIÓN COMPLEMENTARIA

APIs & SDKs Recomendadas

1. Supabase JavaScript Client

- Docs: <https://supabase.com/docs/reference/javascript>
- Funcionalidad: Auth, Realtime, PostgreSQL access

2. Stripe Elements & Payment Intent

- Docs: <https://stripe.com/docs/payments/accept-a-payment>
- PCI Compliance: <https://stripe.com/docs/security/pci-compliance>

3. Scalapay API

- Docs: <https://docs.scalapay.com>
- Sandbox: Acceso a ambiente de prueba

4. Amadeus GDS (para Fase 2)

- Docs: <https://developers.amadeus.com/>
- Endpoints: /v2/shopping/flight-offers

Herramientas Recomendadas para Desarrollo

Herramienta	Propósito	Link
Replit	IDE online + hosting	replit.com
Supabase	Backend + DB	supabase.com
Stripe Dashboard	Pruebas de pagos	dashboard.stripe.com
Chrome DevTools	Performance audit	devtools
Lighthouse	SEO + Accessibility	chrome.google.com/webstore
Git	Control de versiones	git-scm.com

CONCLUSIONES

Este documento proporciona la **arquitectura técnica y estratégica completa** para desarrollar [TripsEuropa.com](https://tripseuropa.com) como una plataforma premium de viajes.

Puntos clave a recordar:

- ✓ **MVP en 2 semanas:** Funcionalidad básica > perfección inicial
- ✓ **Colores corporativos:** Oro (#D4A66A) + Azul marino (#1B2D4D) = identidad fuerte
- ✓ **Mobile-first:** 60%+ del tráfico será desde celulares
- ✓ **Pagos flexibles:** Scalapay/BNPL es el diferenciador clave

- ✓ **Seguridad:** PCI compliance, autenticación robusta, HTTPS siempre
- ✓ **SEO:** Contenido localizado para viajeros colombianos

Próximos pasos:

1. Clonar repo base en Replit
2. Configurar Supabase + Stripe
3. Iniciar Sprint 1 (Landing page)
4. Iterar basado en feedback de usuarios

¡Éxito en el desarrollo! 🚀