💻 LA RESERVA - CONSTANTES Y CONFIGURACIÓN

PARTE 3: Tipos TypeScript

Versión: 1.0
Fecha: Octubre 2025

📄 ARCHIVO: types/index.ts

Ubicación: `src/types/index.ts`

```
// src/types/index.ts

/**
 * LA RESERVA - TIPOS TYPESCRIPT
 *
 * Tipos basados en el esquema de base de datos Supabase
 * y tipos adicionales para el frontend.
 *
 * @version 1.0
 */

// ============================================
// 1. TIPOS DE BASE DE DATOS
// ============================================

/**
 * Cliente en la base de datos
 */
export interface Client {
  id: string;
  name: string;
  email: string;
  phone: string;
  company?: string;
  notes?: string;
  total_events: number;
  total_spent: number;
  created_at: string;
  updated_at: string;
}

/**
 * Cotización solicitada
 */
export interface Quote {
  id: string;
  client_id?: string;
  client_name: string;
  client_email: string;
  client_phone: string;
  event_type: string;
  event_date: string;
  guest_count: number;
  message?: string;
  status: 'new' | 'contacted' | 'quoted' | 'converted' | 'declined';
  estimated_price?: number;
  admin_notes?: string;
  created_at: string;
  updated_at: string;
  contacted_at?: string;
  converted_at?: string;
}

/**
 * Evento confirmado
 */
export interface Event {
  id: string;
  client_id?: string;
  quote_id?: string;
  event_type: string;
  event_date: string;
  event_time?: string;
  guest_count: number;
  venue?: string;
  venue_address?: string;
  venue_district?: string;
  package_id?: string;
  service_ids?: string[];
  total_price: number;
  deposit_paid: number;
  balance_due?: number;
  status: 'pending' | 'confirmed' | 'completed' | 'cancelled';
  notes?: string;
  special_requests?: string;
  cocktails_selected?: string[];
  created_at: string;
  updated_at: string;
  confirmed_at?: string;
  completed_at?: string;
}

/**
 * Imagen de evento (portafolio)
 */
export interface EventImage {
  id: string;
  event_id?: string;
  image_url: string;
  thumbnail_url?: string;
  caption?: string;
  order_index: number;
  created_at: string;
}

/**
 * Servicio ofrecido
 */
export interface Service {
  id: string;
  name: string;
  slug: string;
  description: string;
  long_description?: string;
  price_from: number;
  features: string[];
  icon?: string;
  image_url?: string;
  active: boolean;
  order_index: number;
  created_at: string;
  updated_at: string;
}

/**
 * Paquete predefinido
 */
export interface Package {
  id: string;
  name: string;
  slug: string;
  description: string;
  price: number;
  guest_range: string;
  duration: number;
  features: string[];
  popular: boolean;
  active: boolean;
  order_index: number;
  created_at: string;
  updated_at: string;
}

/**
 * Testimonio de cliente
 */
export interface Testimonial {
  id: string;
  client_name: string;
  client_company?: string;
  event_type?: string;
  rating: number;
  comment: string;
  image_url?: string;
  approved: boolean;
  featured: boolean;
  created_at: string;
}

/**
 * Post del blog
 */
export interface BlogPost {
  id: string;
  title: string;
  slug: string;
  excerpt: string;
  content: string;
  image_url?: string;
  author_id?: string;
  published: boolean;
  published_at?: string;
  views: number;
  created_at: string;
  updated_at: string;
}

/**
 * Usuario administrador
 */
export interface AdminUser {
  id: string;
  email: string;
  role: 'super_admin' | 'admin';
  full_name: string;
  avatar_url?: string;
  created_at: string;
  updated_at: string;
}

/**
 * Configuración del sitio
 */
export interface SiteSetting {
  key: string;
  value: any; // JSON value
  description?: string;
  updated_at: string;
}

// ============================================
// 2. TIPOS DE FORMULARIOS
// ============================================

/**
 * Datos del formulario de cotización
 */
export interface QuoteFormData {
  name: string;
  email: string;
  phone: string;
  eventType: string;
  eventDate: string;
  guestCount: number;
  message?: string;
}

/**
 * Datos del formulario de contacto
 */
export interface ContactFormData {
  name: string;
  email: string;
  phone?: string;
  subject: string;
  message: string;
}

/**
 * Datos del formulario de newsletter
 */
export interface NewsletterFormData {
  email: string;
}

/**
 * Datos del formulario de login admin
 */
export interface LoginFormData {
  email: string;
  password: string;
}

/**
 * Datos del formulario de evento (admin)
 */
export interface EventFormData {
  clientName: string;
  clientEmail: string;
  clientPhone: string;
  eventType: string;
  eventDate: string;
  eventTime?: string;
  guestCount: number;
  venue?: string;
  packageId?: string;
  totalPrice?: number;
  deposit?: number;
  status: 'pending' | 'confirmed' | 'completed' | 'cancelled';
  notes?: string;
}

// ============================================
// 3. TIPOS DE UI/COMPONENTES
// ============================================

/**
 * Props del componente Button
 */
export interface ButtonProps {
  variant?: 'primary' | 'secondary' | 'outline';
  size?: 'sm' | 'md' | 'lg';
  fullWidth?: boolean;
  disabled?: boolean;
  loading?: boolean;
  href?: string;
  onClick?: () => void;
  children: React.ReactNode;
  className?: string;
}

/**
 * Props del componente Card
 */
export interface CardProps {
  hover?: boolean;
  padding?: 'none' | 'sm' | 'md' | 'lg';
  children: React.ReactNode;
  className?: string;
}

/**
 * Props del componente Modal
 */
export interface ModalProps {
  isOpen: boolean;
  onClose: () => void;
  title?: string;
  children: React.ReactNode;
  size?: 'sm' | 'md' | 'lg' | 'xl';
}

/**
 * Props del componente ServiceCard
 */
export interface ServiceCardProps {
  service: Service;
  showDetails?: boolean;
}

/**
 * Props del componente TestimonialCard
 */
export interface TestimonialCardProps {
  testimonial: Testimonial;
}

// ============================================
// 4. TIPOS DE RESPUESTA API
// ============================================

/**
 * Respuesta genérica de API
 */
export interface ApiResponse<T = any> {
  success: boolean;
  data?: T;
  error?: string;
  message?: string;
}

/**
 * Respuesta paginada
 */
export interface PaginatedResponse<T> {
  data: T[];
  total: number;
  page: number;
  pageSize: number;
  totalPages: number;
}
```

```
/**
 * Error de API
 */
export interface ApiError {
  message: string;
  code?: string;
  details?: any;
}

// ============================================
// 5. TIPOS DE UTILIDAD
// ============================================

/**
 * Opciones de select/dropdown
 */
export interface SelectOption {
  value: string;
  label: string;
  icon?: string;
  description?: string;
}

/**
 * Breadcrumb item
 */
export interface BreadcrumbItem {
  label: string;
  href: string;
  active?: boolean;
}

/**
 * Metadata de página (SEO)
 */
export interface PageMetadata {
  title: string;
  description: string;
  keywords?: string[];
  image?: string;
  url?: string;
  noindex?: boolean;
}
```

```
/**
 * Filtros de búsqueda
 */
export interface SearchFilters {
  query?: string;
  category?: string;
  dateFrom?: string;
  dateTo?: string;
  status?: string;
  sortBy?: string;
  sortOrder?: 'asc' | 'desc';
}

/**
 * Estadísticas del dashboard
 */
export interface DashboardStats {
  totalEvents: number;
  upcomingEvents: number;
  pendingQuotes: number;
  totalRevenue: number;
  monthlyRevenue: number;
  averageRating: number;
}

// ============================================
// 6. TIPOS EXTENDIDOS (con relaciones)
// ============================================

/**
 * Evento con información del cliente
 */
export interface EventWithClient extends Event {
  client?: Client;
}

/**
 * Cotización con información del cliente
 */
export interface QuoteWithClient extends Quote {
  client?: Client;
}
```

```
/**
 * Imagen de evento con información del evento
 */
export interface EventImageWithEvent extends EventImage {
  event?: Event;
}

/**
 * Post de blog con información del autor
 */
export interface BlogPostWithAuthor extends BlogPost {
  author?: AdminUser;
}

// ============================================
// 7. TIPOS DE CONSTANTES
// ============================================

/**
 * Tipo para estados de cotización
 */
export type QuoteStatus = keyof typeof import('./constants').QUOTE_STATUS

/**
 * Tipo para estados de evento
 */
export type EventStatus = keyof typeof import('./constants').EVENT_STATUS

/**
 * Tipo para tipos de evento
 */
export type EventType = typeof import('./constants').EVENT_TYPES[number][

/**
 * Tipo para servicios
 */
export type ServiceId = typeof import('./constants').SERVICES[number]['i

/**
 * Tipo para paquetes
 */
export type PackageId = typeof import('./constants').PACKAGES[number]['i

// ============================================
// 8. TIPOS DE SUPABASE (Database)
// ============================================

/**
 * Tipo de base de datos completo de Supabase
 * (Generado automáticamente con: supabase gen types typescript)
 */
export type Database = {
  public: {
    Tables: {
      clients: {
        Row: Client;
        Insert: Omit<Client, 'id' | 'created_at' | 'updated_at'>;
        Update: Partial<Omit<Client, 'id'>>;
      };
      quotes: {
        Row: Quote;
        Insert: Omit<Quote, 'id' | 'created_at' | 'updated_at'>;
        Update: Partial<Omit<Quote, 'id'>>;
      };
      events: {
        Row: Event;
        Insert: Omit<Event, 'id' | 'created_at' | 'updated_at'>;
        Update: Partial<Omit<Event, 'id'>>;
      };
      event_images: {
        Row: EventImage;
        Insert: Omit<EventImage, 'id' | 'created_at'>;
        Update: Partial<Omit<EventImage, 'id'>>;
      };
      services: {
        Row: Service;
        Insert: Omit<Service, 'id' | 'created_at' | 'updated_at'>;
        Update: Partial<Omit<Service, 'id'>>;
      };
      packages: {
        Row: Package;
        Insert: Omit<Package, 'id' | 'created_at' | 'updated_at'>;
        Update: Partial<Omit<Package, 'id'>>;
      };
      testimonials: {
        Row: Testimonial;
        Insert: Omit<Testimonial, 'id' | 'created_at'>;
        Update: Partial<Omit<Testimonial, 'id'>>;
      };
      blog_posts: {
        Row: BlogPost;
        Insert: Omit<BlogPost, 'id' | 'created_at' | 'updated_at'>;
        Update: Partial<Omit<BlogPost, 'id'>>;
      };
      admin_users: {
        Row: AdminUser;
        Insert: Omit<AdminUser, 'id' | 'created_at' | 'updated_at'>;
        Update: Partial<Omit<AdminUser, 'id'>>;
      };
      site_settings: {
        Row: SiteSetting;
        Insert: Omit<SiteSetting, 'updated_at'>;
        Update: Partial<SiteSetting>;
      };
    };
  };
};

/**
 * FIN DE LA PARTE 3
 * Continúa en: 06D-CONSTANTES-Y-CONFIG-Parte4.md
 */
```

✅ VERIFICACIÓN - PARTE 3

- □ Tipos de base de datos coinciden con schema SQL
- □ Tipos de formularios completos
- □ Tipos de componentes UI definidos
- □ Tipos de API response estructurados
- □ Database type de Supabase configurado

Próximo archivo: Parte 4 - Funciones de Utilidad