

Prompt para Backend OpenBlind

Tecnologías Requeridas

- **Node.js con Express.js**
- **PostgreSQL para datos críticos**
- **MongoDB para datos secundarios/metadatos**
- **Arquitectura MVC**
- **Sistema de logging completo**
- **Encriptación para datos sensibles**

Estructura de Base de Datos Híbrida

PostgreSQL (Datos Críticos)

Tabla: users

- **id (PRIMARY KEY, UUID)**
- **email (UNIQUE, ENCRYPTED)**
- **password_hash (ENCRYPTED)**
- **rol (admin/user)**
- **telefono (ENCRYPTED)**
- **nombres (ENCRYPTED)**
- **apellidos (ENCRYPTED)**
- **fecha_nacimiento (ENCRYPTED)**
- **created_at**
- **updated_at**
- **is_active**

Tabla: routes

- **id (PRIMARY KEY, UUID)**
- **name (NOT NULL)**
- **location (NOT NULL)**
- **transport_name (NOT NULL)**
- **status (active/inactive)**

- created_by (FK to users)

- created_at

- updated_at

Tabla: personalized_messages

- id (PRIMARY KEY, UUID)

- message (NOT NULL, ENCRYPTED)

- status (active/inactive)

- route_id (FK to routes)

- created_by (FK to users)

- created_at

- updated_at

Tabla: tourist_registrations

- id (PRIMARY KEY, UUID)

- destination_place (NOT NULL)

- name (NOT NULL)

- description (TEXT)

- latitude (DECIMAL)

- longitude (DECIMAL)

- status (active/inactive)

- created_by (FK to users)

- created_at

- updated_at

MongoDB (Datos Secundarios/Metadatos)

Collection: user_profiles

{

 _id: ObjectId,

 user_id: String, // Referencia a PostgreSQL

 profile_image: String,

```
preferences: Object,  
accessibility_settings: Object,  
last_login: Date,  
login_history: Array,  
device_info: Object,  
location_history: Array,  
created_at: Date,  
updated_at: Date  
}
```

Collection: route_analytics

```
{  
  _id: ObjectId,  
  route_id: String, // Referencia a PostgreSQL  
  usage_statistics: Object,  
  performance_metrics: Object,  
  user_feedback: Array,  
  geographic_data: Object,  
  weather_conditions: Object,  
  traffic_data: Object,  
  created_at: Date,  
  updated_at: Date  
}
```

Collection: voice_guides

```
{  
  _id: ObjectId,  
  route_id: String, // Referencia a PostgreSQL  
  audio_file_url: String,  
  audio_metadata: Object,
```

```
language: String,
voice_settings: Object,
transcription: String,
created_at: Date,
updated_at: Date
}
Collection: system_logs
{
  _id: ObjectId,
  timestamp: Date,
  level: String, // info, warn, error, debug
  method: String,
  url: String,
  status_code: Number,
  request_body: Object,
  response_body: Object,
  user_id: String,
  ip_address: String,
  user_agent: String,
  execution_time: Number,
  error_details: Object
}
```

Estructura de Carpetas Requerida

backend/

└─ src/

| └─ config/

| | └─ database.orm.js // Configuración MongoDB

| | └─ database.sql.js // Configuración PostgreSQL

```
| | └─ encryption.js    // Configuración de encriptación
| | └─ config.js        // Configuraciones generales
| └─ models/
| | └─ sql/             // Modelos PostgreSQL
| | | └─ User.js
| | | └─ Route.js
| | | └─ PersonalizedMessage.js
| | | └─ TouristRegistration.js
| | └─ nosql/           // Modelos MongoDB
| |   └─ UserProfile.js
| |   └─ RouteAnalytics.js
| |   └─ VoiceGuide.js
| |   └─ SystemLog.js
| └─ controllers/
| | └─ authController.js
| | └─ userController.js
| | └─ routeController.js
| | └─ messageController.js
| | └─ touristController.js
| | └─ voiceGuideController.js
| └─ middleware/
| | └─ auth.js
| | └─ validation.js
| | └─ encryption.js
| | └─ logging.js
| | └─ errorHandler.js
| └─ routes/
```

```
| | └─ auth.js
| | └─ users.js
| | └─ routes.js
| | └─ messages.js
| | └─ tourist.js
| | └─ voiceGuides.js
| └─ services/
| | └─ authService.js
| | └─ userService.js
| | └─ routeService.js
| | └─ encryptionService.js
| | └─ logService.js
| | └─ databaseService.js
| └─ utils/
| | └─ constants.js
| | └─ helpers.js
| | └─ validators.js
| └─ app.js
└─ logs/
  └─ application.log // Único archivo de logs
└─ tests/
└─ package.json
└─ .env
└─ .gitignore
└─ server.js
```

Funcionalidades Específicas Requeridas

Sistema de Autenticación

- **JWT para tokens**
- **Bcrypt para hash de contraseñas**
- **Middleware de autenticación**
- **Roles de usuario (admin/user)**

Encriptación de Datos Sensibles

- **AES-256 para encriptar campos sensibles**
- **Variables de entorno para claves de encriptación**
- **Funciones de encriptar/desencriptar en helpers**

Sistema de Logging Completo

- **Winston para logging**
- **Logs categorizados por nivel (info, warn, error, debug)**
- **Middleware que capture todas las peticiones/respuestas**
- **Rotación de logs diaria**
- **Todos los logs almacenados en un solo archivo: application.log**
- **Logs también guardados en MongoDB para consultas avanzadas**

Endpoints Principales Necesarios

// Autenticación

POST /api/auth/login

POST /api/auth/register

POST /api/auth/logout

GET /api/auth/verify-token

// Usuarios

GET /api/users

GET /api/users/:id

PUT /api/users/:id

DELETE /api/users/:id

GET /api/users/profile

// Rutas

GET /api/routes

POST /api/routes

GET /api/routes/:id

PUT /api/routes/:id

DELETE /api/routes/:id

// Mensajes Personalizados

GET /api/messages

POST /api/messages

GET /api/messages/:id

PUT /api/messages/:id

DELETE /api/messages/:id

// Registro Turístico

GET /api/tourist-registrations

POST /api/tourist-registrations

GET /api/tourist-registrations/:id

PUT /api/tourist-registrations/:id

DELETE /api/tourist-registrations/:id

// Guías de Voz

GET /api/voice-guides

POST /api/voice-guides

GET /api/voice-guides/:id

DELETE /api/voice-guides/:id

Validaciones Requeridas

- Validación de email único
- Validación de campos requeridos
- Sanitización de datos de entrada
- Validación de tipos de datos
- Validación de coordenadas geográficas

Manejo de Errores

- Middleware de manejo de errores centralizado
- Códigos de error estandarizados
- Logs detallados de errores
- Respuestas de error consistentes

Variables de Entorno (.env)

Database

POSTGRES_HOST=localhost

POSTGRES_PORT=5432

POSTGRES_DB=openblind_db

POSTGRES_USER=postgres

POSTGRES_PASSWORD=password

MONGODB_URI=mongodb://localhost:27017/openblind_nosql

Security

JWT_SECRET=your-jwt-secret

ENCRYPTION_KEY=your-encryption-key

BCRYPT_ROUNDS=12

Server

PORT=3000

NODE_ENV=development

Logging

LOG_LEVEL=info

LOG_FILE_PATH=./logs/

Consideraciones Especiales

- 1. Accesibilidad:** El backend debe estar optimizado para responder rápidamente a dispositivos móviles
- 2. Geolocalización:** Soporte para coordenadas GPS y cálculos de distancia
- 3. Escalabilidad:** Preparado para manejar múltiples usuarios simultáneos
- 4. Seguridad:** Implementar rate limiting y validación exhaustiva
- 5. CORS:** Configurado para el frontend Angular
- 6. Documentación:** Comentarios en código y documentación de APIs

Archivos de Inicialización

- **Scripts de migración para PostgreSQL**
- **Scripts de inicialización para MongoDB**
- **Seeders con datos de prueba**
- **Docker configuration (opcional)**

Por favor, genera toda la estructura del backend siguiendo estas especificaciones, incluyendo todos los archivos mencionados con código completo y funcional.