Manual Técnico

Sistema de Generación de Exámenes con IA

Equipo de Desarrollo

Versión 1.0

Junio 2025

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Propósito del Sistema

El Sistema de Generación de Exámenes con IA es una plataforma web diseñada para crear automáticamente exámenes personalizados utilizando inteligencia artificial. El sistema permite a los usuarios subir documentos, generar preguntas basadas en el contenido, realizar exámenes con temporizador y recibir retroalimentación detallada.

1.2. Alcance

- Generación automática de exámenes a partir de documentos
- Sistema de autenticación de usuarios
- Interfaz web responsive
- Integración con Google Gemini AI
- Almacenamiento en Supabase
- Retroalimentación personalizada

1.3. Audiencia

Este manual está dirigido a desarrolladores, administradores de sistema y personal técnico responsable del mantenimiento y desarrollo del sistema.

Capítulo 2

Arquitectura del Sistema

2.1. Arquitectura General

El sistema sigue una arquitectura de tres capas:

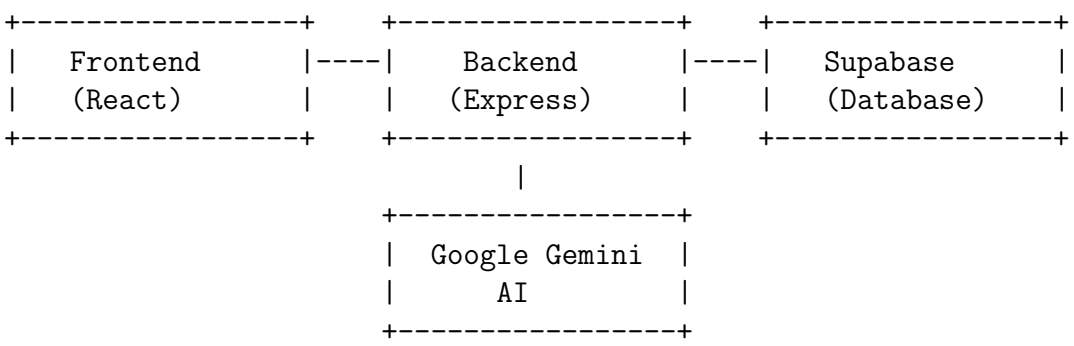

Figura 2.1: Arquitectura del Sistema

2.2. Patrón Arquitectónico

- **Frontend**: SPA (Single Page Application) con React
- **Backend**: API RESTful con Express.js
- **Base de Datos**: PostgreSQL (Supabase)
- **Servicios Externos**: Google Gemini AI para generación de contenido

Capítulo 3

Tecnologías Utilizadas

3.1. Frontend

Tecnología	Versión/Descripción
React	19.0.0 - Framework principal
TypeScript	Tipado estático
Vite	Build tool y dev server
Tailwind CSS	Framework de estilos
React Router	Enrutamiento
Motion	Animaciones
SweetAlert2	Alertas y modales
React Markdown	Renderizado de markdown
KaTeX	Renderizado de fórmulas matemáticas

Cuadro 3.1: Tecnologías del Frontend

3.2. Backend

Tecnología	Descripción
Node.js	Runtime de JavaScript
Express.js	Framework web
Google GenAI	Integración con Gemini AI
Supabase	Base de datos y autenticación
Multer	Manejo de archivos
CORS	Cross-Origin Resource Sharing
dotenv	Variables de entorno

Cuadro 3.2: Tecnologías del Backend

3.3. Base de Datos

- **Supabase (PostgreSQL)**: Base de datos principal
- **Autenticación**: Sistema de usuarios integrado

Capítulo 4

Instalación y Configuración

4.1. Requisitos Previos

- Node.js 18+
- npm o yarn
- Cuenta de Supabase
- API Key de Google Gemini

4.2. Instalación Frontend

```
cd frontend
npm install
```

4.3. Instalación Backend

```
cd backend
npm install
```

4.4. Variables de Entorno

4.4.1. Frontend (.env)

```
VITE_SUPABASE_URL=tu_supabase_url
VITE_SUPABASE_ANON_KEY=tu_supabase_anon_key
VITE_BACKEND_URL=http://localhost:3001
```

4.4.2. Backend (.env)

```
GEMINI_API_KEY=tu_gemini_api_key
SUPABASE_URL=tu_supabase_url
SUPABASE_SERVICE_ROLE_KEY=tu_service_role_key
PORT=3001
```

Capítulo 5

Estructura del Proyecto

5.1. Frontend

```
frontend/
+-- src/
    +-- API/
        +-- Gemini.tsx
    +-- components/
        +-- Main/
        +-- Examen/
        +-- ...
    +-- context/
        +-- AuthContext.tsx
    +-- pages/
        +-- Login.tsx
        +-- Examenes.tsx
        +-- ...
    +-- routers/
        +-- routes.tsx
    +-- ...
+-- public/
+-- package.json
+-- vite.config.ts
```

5.2. Backend

```
backend/
+-- src/
    +-- index.js          # Servidor principal
    +-- reqAuthMiddleware.js
    +-- reqSupabase.js
    +-- local.js
    +-- analize.js
    +-- ...
+-- package.json
+-- supabase.config.js
```

Capítulo 6

Componentes Principales

6.1. Frontend Components

6.1.1. AuthContext

- **Propósito**: Manejo global del estado de autenticación
- **Ubicación**: `src/context/AuthContext.tsx`
- **Funcionalidades**:
  - Login/logout de usuarios
  - Verificación de estado de sesión
  - Protección de rutas

6.1.2. ExamenPage

- **Propósito**: Interfaz principal para realizar exámenes
- **Ubicación**: `src/Examen/ExamenPage.tsx`
- **Funcionalidades**:
  - Renderizado de preguntas
  - Timer de examen
  - Selección de respuestas
  - Envío de resultados

6.1.3. Navbar

- **Propósito**: Navegación principal
- **Ubicación**: `src/components/Navbar.tsx`
- **Funcionalidades**:
  - Enlaces de navegación
  - Estado de autenticación
  - Menú responsivo

6.2. Backend Modules

6.2.1. Authentication Middleware

- **Propósito**: Verificación de tokens de usuario
- **Ubicación**: `src/reqAuthMiddleware.js`
- **Función**: `getUserFromRequest()`

6.2.2. Supabase Integration

- **Propósito**: Operaciones de base de datos
- **Ubicación**: `src/reqSupabase.js`
- **Funciones principales**:
  - `CreateAuthExamUser()`
  - `CreateAuthFeedback()`
  - `verifyAuthExamUser()`

Capítulo 7

API Endpoints

7.1. Generación de Exámenes

7.1.1. POST /api/upload_files

- **Descripción**: Genera examen a partir de archivos subidos
- **Autenticación**: Requerida
- **Body**:

```
{
  "prompt": "string",
  "tiempo_limite_segundos": "number"
}
```

- **Archivos**: Multipart/form-data
- **Respuesta**: Objeto examen creado

7.1.2. POST /api/generate-content

- **Descripción**: Genera examen a partir de prompt de texto
- **Autenticación**: Requerida
- **Body**:

```
{
  "prompt": "string",
  "dificultad": "easy|medium|hard",
  "tiempo_limite_segundos": "number"
}
```

7.1.3. POST /api/generate-content-based-on-history

- **Descripción**: Genera examen basado en historial de usuario
- **Autenticación**: Requerida
- **Body**:

```
{
  "exams_id": ["array_of_exam_ids"],
  "prompt": "string",
  "tiempo_limite_segundos": "number"
}
```

7.2. Retroalimentación

7.2.1. POST /api/generate-feedback

- **Descripción**: Genera retroalimentación personalizada
- **Autenticación**: Requerida
- **Body**:

```
{
  "examen_id": "string"
}
```

Capítulo 8

Base de Datos

8.1. Estructura de Tablas

8.1.1. Tabla: users

```
- id (uuid, primary key)
- email (varchar)
- created_at (timestamp)
- updated_at (timestamp)
```

8.1.2. Tabla: exams

```
- id (uuid, primary key)
- user_id (uuid, foreign key)
- titulo (varchar)
- descripcion (text)
- preguntas (jsonb)
- dificultad (varchar)
- numero_preguntas (integer)
- tiempo_limite (integer)
- created_at (timestamp)
```

8.1.3. Tabla: exam_results

```
- id (uuid, primary key)
- exam_id (uuid, foreign key)
- user_id (uuid, foreign key)
- respuestas (jsonb)
- puntaje (integer)
- tiempo_empleado (integer)
- completed_at (timestamp)
```

8.1.4. Tabla: feedback

```
- id (uuid, primary key)
- exam_id (uuid, foreign key)
- user_id (uuid, foreign key)
```

```
- feedback_data (jsonb)
- created_at (timestamp)
```

Capítulo 9

Integración con IA

9.1. Google Gemini AI

9.1.1. Modelos Utilizados

- **gemini-2.0-flash**: Para retroalimentación y exámenes fáciles
- **gemini-2.5-pro-exp-03-25**: Para exámenes complejos y análisis avanzado

9.1.2. Prompt Engineering

El sistema utiliza instrucciones estructuradas para generar contenido consistente:

```
const systemInstruction = `
Analiza el contenido y crea un examen con la siguiente estructura JSON:
{
  "dato": [
    {
      "id": 1,
      "pregunta": "Pregunta del examen",
      "opciones": ["opcion1", "opcion2", "opcion3", "opcion4"],
      "correcta": 0
    }
  ],
  "titulo": "Titulo del examen",
  "numero_preguntas": 5,
  "descripcion": "Descripcion del contenido",
  "dificultad": "easy|medium|hard|mixed"
}
`;
```

9.1.3. Procesamiento de Archivos

- Soporte para múltiples formatos de documento
- Extracción de contenido para análisis
- Limpieza automática de archivos temporales

Capítulo 10

Funcionalidades del Sistema

10.1. Gestión de Usuarios

- Registro y autenticación
- Perfiles de usuario
- Historial de exámenes
- Estadísticas personalizadas

10.2. Generación de Exámenes

- A partir de documentos subidos
- Basado en prompts de texto
- Utilizando historial previo
- Configuración de dificultad y tiempo

10.3. Realización de Exámenes

- Interfaz intuitiva de preguntas
- Timer configurable
- Navegación entre preguntas
- Guardado automático de progreso

10.4. Retroalimentación

- Análisis detallado por pregunta
- Explicaciones de respuestas correctas
- Identificación de áreas de mejora

- Retroalimentación personalizada con IA

10.5. Historial y Estadísticas

- Registro de exámenes completados
- Métricas de rendimiento
- Análisis de progreso
- Logros y metas

Capítulo 11

Configuración de Entorno

11.1. Desarrollo

11.1.1. Frontend

```
cd frontend
npm run dev
```

- Puerto: 5173 (Vite default)
- Hot reload activado
- DevTools habilitadas

11.1.2. Backend

```
cd backend
node src/index.js
```

- Puerto: 3001
- CORS configurado para desarrollo

11.2. Producción

11.2.1. Build Frontend

```
cd frontend
npm run build
```

11.2.2. Variables de Entorno de Producción

- Configurar URLs de produccion
- Usar HTTPS
- Configurar CORS específico
- Habilitar logs de produccion

Capítulo 12

Despliegue

12.1. Frontend (Hostinger)

```
npm run build
# Subir carpeta dist/ al hosting
```

12.2. Backend

```
# En servidor de produccion
git clone [repositorio]
cd backend
npm install --production
pm2 start src/index.js --name "exam-backend"
```

12.3. Base de Datos

- Configurar Supabase en produccion
- Migrar esquemas si es necesario
- Configurar backups automáticos

Capítulo 13

Mantenimiento

13.1. Logs

- Revisar logs de aplicacion regularmente
- Monitorear errores de IA
- Verificar uso de cuotas de API

13.2. Base de Datos

- Backup regular de datos
- Limpieza de archivos temporales
- Optimización de consultas

13.3. Actualizaciones

- Dependencias de Node.js
- Versiones de React
- APIs de Google Gemini

Capítulo 14

Solución de Problemas

14.1. Errores Comunes

14.1.1. Error de API Key de Gemini

```
Error: Falta la variable de entorno GEMINI_API_KEY
```

Solución: Verificar que la variable esté configurada en `.env`

14.1.2. Error de CORS

```
Access to fetch blocked by CORS policy
```

Solución: Verificar configuracion de CORS en backend

14.1.3. Error de Autenticación Supabase

```
Invalid JWT token
```

Solución: Verificar configuracion de Supabase y tokens

14.2. Monitoreo

14.2.1. Métricas Importantes

- Tiempo de respuesta de API
- Tasa de errores de generación
- Uso de cuota de Gemini AI
- Carga de base de datos

14.2.2. Herramientas Recomendadas

- Logs de aplicacion
- Métricas de Supabase
- Google Cloud Console (para Gemini)

Capítulo 15

Conclusión

Este manual técnico proporciona una guía completa para la comprensión, instalación, configuracion y mantenimiento del Sistema de Generación de Exámenes con IA. Para actualizaciones y modificaciones, consultar el repositorio del proyecto y mantener actualizada la documentación.

Versión del Manual: 1.0
Fecha: Junio 2025
Desarrollado por: Equipo de Desarrollo