



Solución: Base de Datos Vacía



Problema Identificado

La aplicación **English Master Pro** estaba funcionando correctamente, pero la base de datos estaba completamente vacía. Los usuarios podían registrarse e iniciar sesión, pero no había contenido disponible:

- ❌ Sin verbos en el sistema
- ❌ Sin palabras del diccionario
- ❌ Sin lecciones
- ❌ Sin juegos configurados
- ❌ Sin logros



Solución Implementada



Descubrimiento

Al analizar el proyecto, descubrimos que **ya existe** un script de seed completo con datos educativos de alta calidad:

- 📍 **Ubicación:** `scripts/seed.ts`
- 📊 **Datos incluidos:**
 - **1,000 verbos** con conjugaciones completas
 - **1,100 palabras** organizadas por niveles CEFR (A1-C2)
 - **5 juegos** educativos
 - **8 logros** para motivación
 - **5 lecciones** de ejemplo
- **Usuario de prueba** (john@doe.com / johndoe123)



Contenido de los Datos

1. Verbos (1,000 verbos)

Cada verbo incluye:

- Infinitivo y todas las conjugaciones (3ra persona, gerundio, pasado simple, participio)
- Traducción al español
- Pronunciación IPA
- URL de audio (cuando disponible)
- Nivel CEFR (A1-C2)
- Categoría (existence, action, communication, etc.)
- Propiedades (irregular, modal, phrasal)
- Ejemplos de uso

Ejemplo de verbo:

```
{
  "infinitive": "be",
  "conjugations": {
    "third_person_singular": "is",
    "present_participle": "being",
    "simple_past": "was",
    "past_participle": "were"
  },
  "spanish_translation": "ser/estar",
  "pronunciation": {
    "ipa": "/ 'bi/, /bi/",
    "audio_url": "https://ssl.gstatic.com/dictionary/static/sounds/20220808/be--_us_1.mp3"
  },
  "cefr_level": "A1",
  "category": "existence",
  "properties": {
    "irregular": true,
    "modal": false,
    "phrasal": false
  }
}
```

2. Diccionario (1,100 palabras)

Palabras organizadas por niveles CEFR:

- **A1**: 178 palabras (16.2%)
- **A2**: 192 palabras (17.5%)
- **B1**: 179 palabras (16.3%)
- **B2**: 178 palabras (16.2%)
- **C1**: 186 palabras (16.9%)
- **C2**: 187 palabras (17.0%)

Cada palabra incluye:

- Inglés y español
- Pronunciación
- Parte del discurso (noun, verb, adjective, etc.)
- Definición
- Ejemplos de uso
- Nivel de dificultad
- Categoría (general, travel, business, academic, etc.)

3. Juegos Educativos (5 juegos)

- **Vocabulary Quiz**: Preguntas de opción múltiple (10 XP)
- **Memory Match**: Emparejar palabras en inglés y español (15 XP)
- **Typing Challenge**: Escribir traducciones rápidamente (8 XP)
- **Pronunciation Practice**: Práctica con reconocimiento de voz (12 XP)
- **Verb Conjugation**: Practicar conjugaciones verbales (15 XP)

4. Logros (8 achievements)

- **First Steps**: Completar primera lección (25 XP)
- **Word Explorer**: Aprender 50 palabras (100 XP)
- **Streak Master**: Mantener racha de 7 días (200 XP)
- **Quiz Champion**: Obtener 100% en 10 quizzes (150 XP)

- **Vocabulary Genius:** Aprender 500 palabras (500 XP)
- **XP Collector:** Ganar 1000 XP total (100 XP)
- **Pronunciation Pro:** 20 pronunciaciones perfectas (300 XP)
- **English Master:** Alcanzar nivel C2 (1000 XP)

5. Lecciones de Ejemplo (5 lecciones)

- **Basic Greetings** (A1): Saludos básicos
- **Family Members** (A1): Vocabulario familiar
- **Present Simple Tense** (A2): Tiempo presente simple
- **Business Vocabulary** (B2): Vocabulario de negocios
- **Advanced Conversation** (C1): Conversación avanzada



Cómo Ejecutar el Seed en Vercel

Opción 1: Desde la Terminal de Vercel (Recomendado)

1. Ir al Dashboard de Vercel:

- Visita: <https://vercel.com/dashboard>
- Selecciona tu proyecto "english-master-pro"

2. Abrir la Terminal:

- Ve a la pestaña **Settings**
- Busca la sección **Functions**
- Abre una terminal SSH (si está disponible)

3. Ejecutar el seed:

```
bash
```

```
npm run seed
```

Opción 2: Usando Vercel CLI

Si tienes Vercel CLI instalado en tu computadora:

```
# Instalar Vercel CLI (si no lo tienes)
npm install -g vercel

# Iniciar sesión
vercel login

# Navegar a tu proyecto local
cd /ruta/a/tu/proyecto

# Ejecutar el seed con las variables de entorno de producción
vercel env pull .env.production
npx tsx --require dotenv/config -r dotenv/config scripts/seed.ts
```

Opción 3: Crear un Script de Despliegue

Puedes crear un endpoint API temporal en tu aplicación para ejecutar el seed:

Crear: `app/api/seed/route.ts`

```
import { NextResponse } from 'next/server';
import { exec } from 'child_process';
import { promisify } from 'util';

const execPromise = promisify(exec);

// IMPORTANTE: Protege este endpoint con una clave secreta
export async function POST(request: Request) {
  try {
    const { secret } = await request.json();

    // Verificar clave secreta
    if (secret !== process.env.SEED_SECRET) {
      return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
    }

    // Ejecutar seed
    const { stdout, stderr } = await execPromise('npm run seed');

    return NextResponse.json({
      success: true,
      output: stdout,
      errors: stderr
    });
  } catch (error) {
    return NextResponse.json({
      success: false,
      error: error.message
    }, { status: 500 });
  }
}
```

Luego, ejecuta:

```
curl -X POST https://tu-app.vercel.app/api/seed \
-H "Content-Type: application/json" \
-d '{"secret":"TU_CLAVE_SECRETA"}'
```

⚠ IMPORTANTE: Elimina este endpoint después de ejecutar el seed.

Opción 4: Ejecutar Localmente Conectado a la BD de Producción

```
# 1. Obtener la DATABASE_URL de Vercel
# Ve a Settings > Environment Variables > DATABASE_URL

# 2. Crear archivo .env local
echo "DATABASE_URL=tu_url_de_base_de_datos_de_produccion" > .env

# 3. Ejecutar el seed
npm run seed

# 4. ¡IMPORTANTE! Eliminar el archivo .env después
rm .env
```



Comando de Seed

El comando ya está configurado en `package.json` :

```
"prisma": {
  "seed": "tsx --require dotenv/config scripts/seed.ts"
}
```





Simplemente ejecuta:

```
npm run seed
```

Configuración Técnica

Dependencias Necesarias

Ya están instaladas en el proyecto:

-  `@prisma/client` : Cliente de Prisma
-  `bcryptjs` : Para hashear contraseñas
-  `tsx` : Para ejecutar TypeScript
-  `dotenv` : Para variables de entorno

Variables de Entorno Requeridas

- `DATABASE_URL` : URL de conexión a PostgreSQL (ya configurada en Vercel)

Estructura de Archivos

```
english-master-pro/
├── scripts/
│   └── seed.ts           Script de seed
├── data/
│   ├── verbs_data.json  1,000 verbos
│   ├── dictionary_data.json 1,100 palabras
│   └── ...               Otros archivos de datos
├── prisma/
│   └── schema.prisma     Esquema de base de datos
└── package.json          Configuración del seed
```

Verificar que el Seed Funcionó

Después de ejecutar el seed, verifica que los datos se cargaron correctamente:






1. Usuario de Prueba

Intenta iniciar sesión con:

- **Email:** john@doe.com
- **Contraseña:** johndoe123

2. Verificar Contenido

En tu aplicación, verifica que ahora aparezcan:

-  Verbos en la sección de verbos
-  Palabras en el diccionario
-  Lecciones disponibles
-  Juegos funcionando
-  Logros visibles

3. Consultar Directamente la Base de Datos

Si tienes acceso a la base de datos:

```
-- Contar verbos
SELECT COUNT(*) FROM "Verb";

-- Contar palabras
SELECT COUNT(*) FROM "Word";

-- Contar lecciones
SELECT COUNT(*) FROM "Lesson";

-- Contar juegos
SELECT COUNT(*) FROM "Game";

-- Contar logros
SELECT COUNT(*) FROM "Achievement";
```

Deberías ver:

- Verbs: 1000
- Words: 1100
- Lessons: 5
- Games: 5
- Achievements: 8



Resultado Esperado

Una vez ejecutado el seed, tu aplicación tendrá:

- ☒ **1,000 verbos** con conjugaciones completas
- ☒ **1,100 palabras** organizadas por niveles CEFR
- ☒ **5 juegos educativos** interactivos
- ☒ **8 logros** para motivar a los usuarios
- ☒ **5 lecciones** de ejemplo
- ☒ **Usuario de prueba** para testing



Solución de Problemas

Error: “Cannot find module ‘bcryptjs’”

```
npm install bcryptjs @types/bcryptjs
```

Error: “Cannot find module ‘@prisma/client’”

```
npm install @prisma/client
npx prisma generate
```

Error: “ECONNREFUSED” o problemas de conexión

- Verifica que la variable `DATABASE_URL` esté correctamente configurada
- Asegúrate de que la base de datos esté accesible

- Verifica que el formato de la URL sea correcto: `postgresql://user:password@host:port/database`

Error: “Unique constraint failed”

Si el seed ya se ejecutó antes, puedes:

1. **Limpiar la base de datos** (⚠️ esto borrará todos los datos):

```
bash
```

```
npx prisma db push --force-reset
```

```
npm run seed
```

1. **O ejecutar solo partes del seed** modificando el script



Recursos Adicionales

- **Prisma Seed Documentation:** <https://www.prisma.io/docs/guides/database/seed-database>
- **Vercel Environment Variables:** <https://vercel.com/docs/concepts/projects/environment-variables>
- **Vercel CLI Documentation:** <https://vercel.com/docs/cli>



Checklist de Verificación

- ☐ Script de seed está en `scripts/seed.ts`
- ☐ Archivos de datos están en `data/`
- ☐ Comando seed configurado en `package.json`
- ☐ Dependencias instaladas (tsx, bcryptjs)
- ☐ Variable DATABASE_URL configurada
- ☐ Seed ejecutado exitosamente
- ☐ Datos verificados en la aplicación
- ☐ Usuario de prueba funciona



Seguridad

IMPORTANTE:

- ⚠️ Si creas un endpoint API para el seed, protégelo con una clave secreta
- ⚠️ Elimina el endpoint después de usarlo
- ⚠️ No expongas la DATABASE_URL en el código o en repositorios públicos
- ⚠️ El seed crea un usuario de prueba (john@doe.com) - considera eliminarlo en producción



Contacto y Soporte

Si tienes problemas ejecutando el seed:

1. Revisa los logs de Vercel
2. Verifica las variables de entorno
3. Asegúrate de que la base de datos esté activa
4. Consulta la documentación de Prisma

Autor: DeepAgent - Abacus.AI
Fecha: Octubre 2025
Versión: 1.0