

Informe Técnico de Despliegue:

Proyecto Codexy

Fecha del Informe: 07 de Diciembre de 2025

Entorno: Producción (Azure Cloud)

Suscripción: Azure for Students

1. Resumen Ejecutivo

El proyecto "Codexy", compuesto por una API REST en .NET, una base de datos SQL Server, un frontend web en Angular y una aplicación móvil híbrida en Ionic/Capacitor, ha sido migrado exitosamente desde un entorno local de desarrollo a una infraestructura de nube en Microsoft Azure. Se priorizó el uso de servicios gestionados bajo la capa gratuita (Free Tier) para optimizar costos y rendimiento.

2. Arquitectura de Infraestructura (Azure)

Se ha establecido una arquitectura PaaS (Platform as a Service) para garantizar escalabilidad y reducir la carga operativa.

Componente	Recurso en Azure	Configuración / SKU	Región
Backend API	Azure App Service (Linux)	Plan F1 (Gratuito)	East US 2 / South Central US
Base de Datos	Azure SQL Database	Nivel Básico (Basic), 2GB, 5 DTU	East US 2 / South Central US
Frontend Web	Azure Static Web Apps	Plan Free (Gratuito)	East US 2
Tiempo Real	Azure SignalR Service	Tier Free_F1 (Default)	East US 2

3. Detalles del Despliegue por Componente

3.1. Base de Datos (Azure SQL)

- **Creación:** Se provisionó un servidor SQL y una base de datos única. Debido a restricciones de política en la suscripción educativa para la región "Brazil South" y capacidad en "East US", se optó por regiones alternativas permitidas como East US 2 o South Central US .
- **Migración de Esquema:** Se ejecutaron las migraciones de Entity Framework Core desde el entorno local (dotnet ef database update) apuntando temporalmente a la cadena de conexión de Azure .
- **Sembrado de Datos (Seed Data):** Se detectó que la lógica de seed estaba restringida al entorno de desarrollo. Se ejecutó una carga inicial manual conectando la API local (en modo Dev) a la base de datos productiva .

3.2. Backend API (.NET 9 / App Service)

- **Configuración:** Se migraron los secretos de usuario (user-secrets) a las "Variables de Entorno" (Application Settings) del App Service. Esto incluyó cadenas de conexión, credenciales JWT, Cloudinary y SMTP .
- **Sistema Operativo:** Se seleccionó Linux por eficiencia y compatibilidad nativa con .NET Core/8 .
- **Publicación:** Se utilizó un perfil de publicación directo desde Visual Studio autenticado con la cuenta de Azure .
- **Ajustes de CORS:** Se configuró una política robusta para permitir orígenes múltiples (Web, Localhost, Apps Móviles) .

3.3. Frontend Web (Angular)

- **CI/CD:** Se implementó integración continua mediante **GitHub Actions**, conectando el repositorio a Azure Static Web Apps .
- **Configuración de Build:**
 - *App Location:* Frontend (Raíz del proyecto Angular) .
 - *Output Location:* dist/frontend/browser (Ruta específica para Angular moderno) .
- **Enrutamiento SPA:** Se creó el archivo staticwebapp.config.json para gestionar el "fallback" al index.html y evitar errores 404 al recargar páginas .

3.4. Tiempo Real (WebSockets / SignalR)

- **Cambio Arquitectónico:** Inicialmente se planeó usar Redis como *Backplane*. Sin embargo, se migró a **Azure SignalR Service** por recomendación de experto, debido a limitaciones de conexiones WebSocket en el App Service (Plan F1) y para reducir costos (\$16 USD vs Gratis) .
- **Implementación:** Se sustituyó la librería StackExchange.Redis por

- Microsoft.Azure.SignalR en el backend .
 - **Autenticación:** Se estandarizó el cliente Angular para usar accessTokenFactory en lugar de cookies, garantizando compatibilidad con el servicio gestionado de SignalR .
-

4. Retos Técnicos y Soluciones (Troubleshooting Log)

Durante el proceso de despliegue se identificaron y resolvieron las siguientes incidencias críticas:

1. **Restricciones de Región (Policy Error):**
 - *Error:* RequestDisallowedByAzure al intentar crear recursos en Brazil South o East US.
 - *Solución:* Identificación de regiones permitidas en la "lista blanca" de la suscripción educativa (East US 2 / South Central US) .
2. **Ruta de Build en CI/CD:**
 - *Error:* App Directory Location: 'Codexy_Project/Frontend' is invalid.
 - *Solución:* Corrección del archivo YAML del workflow para apuntar directamente a la carpeta Frontend en la raíz del repositorio .
3. **Error de Validación en Configuración SPA:**
 - *Error:* Uso de doble comodín (**) en staticwebapp.config.json.
 - *Solución:* Corrección de sintaxis a un solo asterisco (*) en la propiedad de exclusión .
4. **CORS en Aplicación Móvil (Ionic/Capacitor):**
 - *Problema:* Bloqueo de peticiones desde el dispositivo móvil o emulador.
 - *Análisis:* El origen enviado por el WebView a menudo es http://localhost (sin puerto) o capacitor://.
 - *Solución:* Actualización de la lista de orígenes permitidos en Azure App Service para incluir explícitamente:
 - http://localhost
 - capacitor://localhost
 - ionic://localhost
 - null .

5. Estado Final y Recomendaciones

Estado Actual

El sistema se encuentra **100% operativo en producción.**

- **API:** Responde correctamente y documentada con Swagger .
- **Datos:** Persistencia asegurada en SQL Azure con migraciones aplicadas .
- **Cliente Web:** Accesible públicamente con redirección SPA funcional .
- **Cliente Móvil:** Configurado para conectar a la API de producción mediante HTTPS .

Recomendaciones Futuras (Post-Deployment)

1. **Seguridad de Secretos:** Migrar las credenciales almacenadas en *App Service Configuration* hacia un **Azure Key Vault** para mayor seguridad .
 2. **Servicio de Correo:** Reemplazar el uso de SMTP de Gmail por un servicio dedicado como Azure Communication Services o SendGrid para evitar bloqueos y mejorar la entregabilidad .
 3. **Dominio Personalizado:** Configurar un dominio propio (ej. codexy.com) en la Static Web App para reemplazar la URL predeterminada de Azure .
-