

AUTOMATIZACIÓN DE INFRAESTRUCTURA

con Pulumi en el Proyecto MateoTutor

¿Qué es MateoTutor?

Desarrollado en la UACH bajo el proyecto Fondecyt 11220709, Mateo es un tutor virtual que apoya a estudiantes de primer año en cursos de álgebra con alta tasa de reprobación.

Permite seleccionar tópicos clave (factorización, ecuaciones, fracciones, etc.) y ofrece ejercicios paso a paso.

Adapta el contenido según el desempeño del estudiante, reforzando habilidades débiles y promoviendo un aprendizaje personalizado y efectivo.



¿Cuál fue nuestra propuesta?

Una interfaz intuitiva y visualmente atractiva para crear ejercicios sin requerir conocimientos de programación.

Evita el uso de formatos técnicos (como JSON), permitiendo a cualquier usuario estructurar ejercicios fácilmente.

Busca mejorar la experiencia, agilizar el proceso de creación de contenido y fomentar una participación más amplia.

¿Qué es Infraestructura como Código (IaC)?

- Definir y gestionar entornos de infraestructura (servidores, redes, servicios) mediante código reutilizable y versionable.
- Ventajas clave:
 - Reproducibilidad: cualquier miembro del equipo puede levantar el entorno de desarrollo sin errores.
 - Automatización: elimina tareas manuales repetitivas.
 - Control de versiones: el estado de la infraestructura se versiona junto al código.
- Herramientas populares: Terraform, Ansible, y Pulumi

¿Cómo lo usamos en nuestro proyecto?

- **Contexto del Proyecto**
 - **Proyecto:** MateoTutor – Frontend
 - **Tecnologías:** React + TypeScript + Vite
 - **Módulo específico:** newExercise.txt para creación y edición de ejercicios.
- **Necesidades técnicas**
 - **Levantar rápidamente entornos de desarrollo idénticos.**
 - **Ejecutar el frontend con todas las dependencias y configuraciones listas.**
 - **Evitar configuración manual en cada equipo.**
 -
- **Solución:** Usar Pulumi, ya que permite escribir infraestructura como código en TypeScript, el mismo lenguaje del proyecto, facilitando integración y entendimiento.

¿Por qué usamos Pulumi?

- **¿Por qué Pulumi y no Terraform o Ansible?**
 - Usamos TypeScript, y Pulumi nos permite escribir infraestructura en el mismo lenguaje.
 - Es programable: usamos lógica, condiciones, bucles y funciones como en cualquier app.
 - Evitamos aprender HCL o YAML y mantenemos coherencia en el equipo.

Demo

Recomendaciones y conclusiones

Recomendaciones para otros proyectos:

- Si tu stack usa TypeScript/JavaScript: Pulumi es una excelente opción.
- Usa Docker junto a IaC para lograr entornos portables.

Conclusiones:

- Pulumi nos permitió automatizar entornos de desarrollo sin aprender un lenguaje nuevo.
- Al usar TypeScript, se facilitó la integración con el equipo y el proyecto.

¿Qué podríamos automatizar después?

- Despliegue en la nube (por ejemplo, AWS o GCP) usando el mismo código Pulumi.
 - Variables de entorno seguras con Pulumi Config o Pulumi Secrets.
 - CI/CD: Automatizar pulumi up desde GitHub Actions al hacer push a rama main.
 - Entornos por rama o por feature: deployment previews para QA.
- 