Informe Técnico – Monolithic vs Cloud Native Apps

# 1. Integrantes

- Nombre: Horacio David Bareiro Palacios  
- CI: 4757679  
- Email: [davidbareiro11@fpuna.edu.py](mailto:davidbareiro11@fpuna.edu.py)

- Nombre: René David Acosta Aveiro  
- CI: 5540277  
- Email: reneacostaa99@fpuna.edu.py

# 2. Descripción del Proyecto

Nombre del Proyecto: Tekovia Guasu – Calendario Cultural Nacional  
  
Este proyecto consiste en el desarrollo de una plataforma web para la planificación y gestión del calendario anual de eventos culturales del Paraguay. Implementada con el patrón Modelo-Vista-Controlador (MVC), la plataforma permite que un superusuario por municipio administre eventos culturales como conciertos, festivales folclóricos, exposiciones, entre otros.  
  
La aplicación apunta a facilitar la visibilidad, organización y difusión de actividades culturales, iniciando por la ciudad de Asunción.

# 3. Funcionalidad Desarrollada

La funcionalidad clave desarrollada es la gestión de eventos, que incluye:  
  
- Crear, editar y eliminar eventos culturales.  
- Publicar eventos para visualización ciudadana.  
- Controlar fechas, lugares y descripciones.  
  
La lógica de negocio fue implementada tanto en una aplicación monolítica como en una versión nativa para la nube, asegurando equivalencia funcional.

# 4. Arquitecturas Implementadas

## A. Aplicación Monolítica

- Lenguaje y Frameworks: Python 3.11 con Reflex (Pynecone).  
- Base de datos: SQLite (local) y posterior postgresql.  
- Estructura: Aplicación todo-en-uno que maneja frontend, backend y almacenamiento desde una misma instancia de Reflex.  
- Ventajas: Desarrollo ágil, despliegue inmediato, menos complejidad inicial.

## B. Aplicación Nativa para la Nube

- Servicios separados:  
 - Servicio 1: Aplicación Reflex (backend + frontend).  
 - Servicio 2: Base de datos PostgreSQL.  
- Contenedores: Docker.  
- Orquestador: Docker Compose.  
- Ventajas: Modularidad, portabilidad, escalabilidad horizontal.

# 5. Herramientas Utilizadas

- Reflex (https://reflex.dev)  
- Python 3.11  
- Docker y Docker Compose  
- PostgreSQL  
- GitHub  
- VSCode

# 6. Comparación entre Monolítica y Nativa en la Nube

| Aspecto | Monolítica (Reflex local) | Nativa para la Nube (Docker + PostgreSQL) |
| --- | --- | --- |
| Despliegue | Simple (reflex run) | Más complejo (requiere configuración Docker) |
| Escalabilidad | Limitada | Alta (cada componente independiente) |
| Flexibilidad en cambios | Limitada | Mayor flexibilidad en reemplazo de servicios |
| Persistencia de datos | Local (SQLite) | Persistente (volumen Docker + PostgreSQL) |
| Modularidad | Baja | Alta |

# 7. Diagramas de Arquitectura

Diagrama de arquitectura monolítica (Reflex):

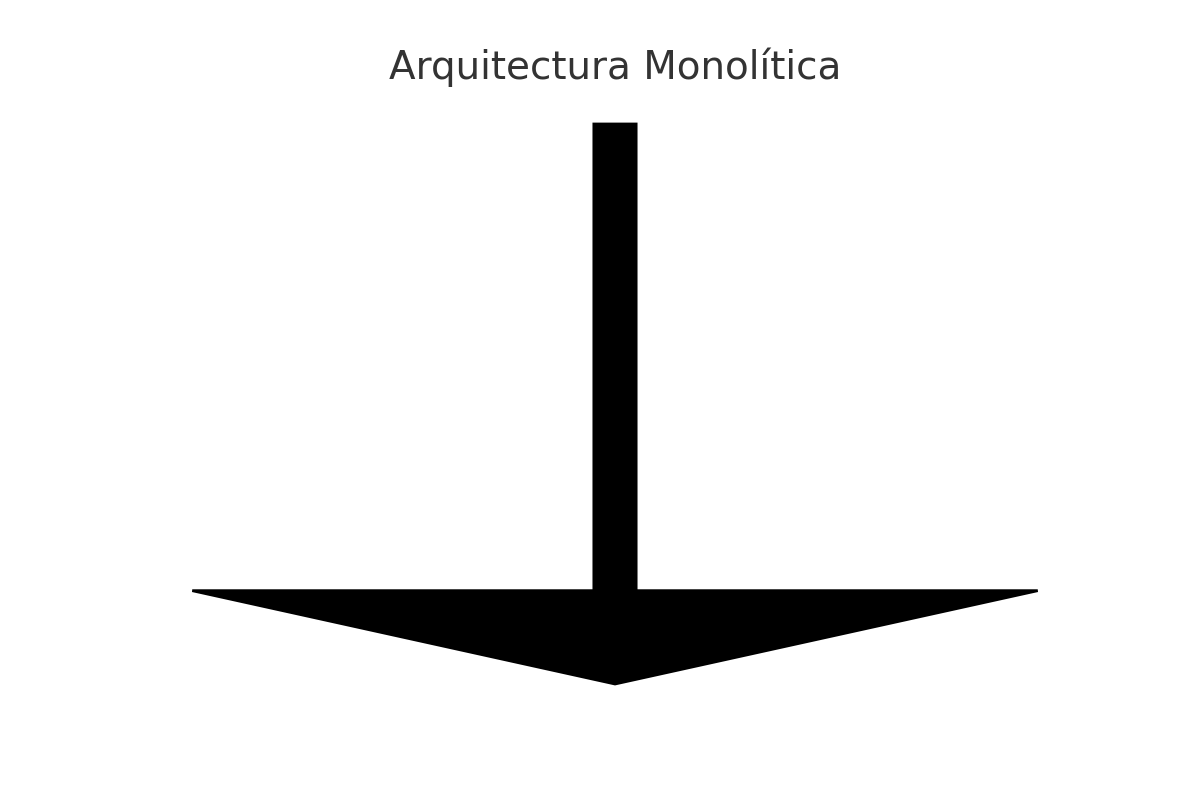
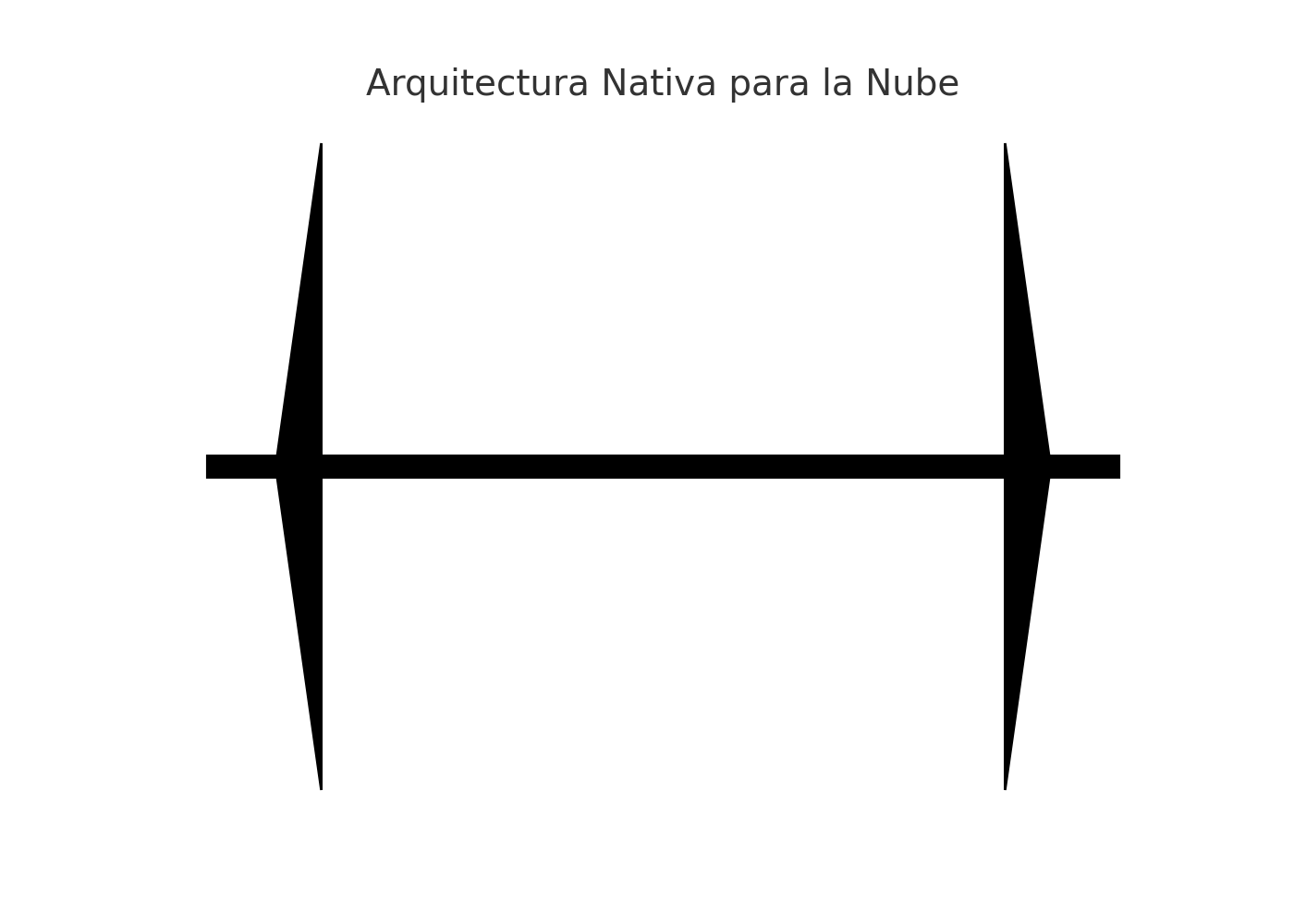


Diagrama de arquitectura nativa para la nube (Docker Compose):



# 8. Instrucciones de Ejecución Local

Monolítica:

git clone https://github.com/davidba11/TekoviaGuasu  
cd TekoviaGuasu  
reflex init  
reflex run

Nativa para la Nube:

docker-compose up --build

# 9. Capturas de Pantalla

\*(Agregar capturas que muestren: creación de evento, listado, edición, etc., tanto en local como vía Docker.)\*

# 10. URL del Repositorio

https://github.com/davidba11/TekoviaGuasu