



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

**Propuesta del Proyecto Apis y Funciones
Jarro_Valle**

Curso: Tópicos de Base de Datos Avanzados

Docente: Mag. Patrick Cuadros

Integrantes:

Jose Luis Jarro Cachi (2020067148)

Gustavo Alonso Valle Bustamante (2020066916)

**Tacna – Perú
2024**

Proyecto

Apis y Funciones Jarro_Valle, Tacna, 2024

Presentado por:

Jose Luis Jarro Cachi

Gustavo Alonso Valle Bustamante

Estudiantes de la Universidad Privada de Tacna

24/11/2024

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	GVB	JJC	JJC	24/11/2024	Versión Original

Contenido

RESUMEN EJECUTIVO.....	4
I Propuesta narrativa.....	5
1. Planteamiento del Problema	5
2. Justificación del proyecto	5
3. Objetivo general	5
4. Beneficios	5
5. Alcance	5
6. Requerimientos del sistema	5
7. Restricciones	6
8. Supuestos.....	6
9. Resultados esperados	6
10. Metodología de implementación	6
11. Actores claves.....	6
12. Papel y responsabilidades del personal	6
13. Plan de monitoreo y evaluación	7
14. Cronograma del proyecto.....	7
15. Hitos de entregables	7
II Presupuesto.....	8
Anexo 01 – Requerimientos del Sistema	9

RESUMEN EJECUTIVO

Nombre del Proyecto propuesto: <i>Proyecto APIs y Funciones Jarro_Valle, Tacna, 2024</i>	
Propósito del Proyecto y Resultados esperados: El propósito del proyecto es desarrollar una API robusta y escalable para la gestión de eventos académicos en instituciones educativas, permitiendo realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre los eventos de manera eficiente. Los resultados esperados son: <ul style="list-style-type: none">- Automatización de la gestión de eventos académicos.- Reducción de errores operativos y mejora en la organización de datos.- Implementación de una plataforma accesible y escalable para los usuarios finales. Aumento en la eficiencia administrativa y toma de decisiones basadas en datos centralizados.	
Población Objetivo: <i>Administradores, docentes y estudiantes de instituciones educativas.</i>	
Monto de Inversión (En Soles): <i>S/. 9063</i>	Monto de Inversión (En Soles): <i>S/. 9063</i>

I Propuesta narrativa

1. Planteamiento del Problema

Las instituciones académicas enfrentan desafíos en la gestión de eventos debido a la falta de herramientas centralizadas y automatizadas. Esto incluye registros dispersos, comunicación ineficaz y un alto margen de errores en procesos manuales. Actualmente, la gestión de eventos depende de métodos obsoletos que no permiten escalabilidad ni integración con otros sistemas.

2. Justificación del proyecto

El proyecto es necesario para optimizar la administración de eventos académicos mediante una solución moderna que automatice procesos, centralice información y permita acceso en tiempo real. Esto beneficiará a los usuarios finales, mejorará la toma de decisiones y reducirá costos operativos, alineándose con las necesidades de digitalización.

3. Objetivo general

Desarrollar e implementar una API que facilite la creación, actualización, consulta y eliminación de eventos académicos, integrándose de manera eficiente en los flujos de trabajo de las instituciones educativas.

4. Beneficios

Tangible:

- Reducción del tiempo de gestión de eventos en un 40%.
- Menor dependencia de procesos manuales y herramientas fragmentadas.
- Incremento en la precisión y actualización de los datos.

Intangible:

- Mejora en la satisfacción del personal administrativo y los usuarios finales.
- Posicionamiento tecnológico de las instituciones educativas.
- Mayor confianza en procesos automatizados.

5. Alcance

- Desarrollo de una API para realizar operaciones CRUD sobre eventos.
- Integración con bases de datos NoSQL (MongoDB).
- Despliegue en contenedores Docker.
- Documentación interactiva con Swagger UI.
- Automatización de pruebas y despliegues mediante GitHub Actions.

6. Requerimientos del sistema

Hardware: Servidores con CPU de 4 núcleos, 8 GB de RAM y 50 GB de almacenamiento SSD.

Software:

- .NET Core para backend.
- MongoDB como base de datos.
- Docker para contenedores.

- GitHub Actions para CI/CD.

7. Restricciones

- Uso obligatorio de MongoDB como base de datos.
- Despliegue en contenedores Docker.
- Tiempo promedio de respuesta <500 ms.

8. Supuestos

- La infraestructura de la institución soportará Docker y MongoDB.
- Los usuarios finales tendrán acceso a dispositivos con conexión a internet.
- La API será integrada únicamente con sistemas compatibles.

9. Resultados esperados

- Incremento en la eficiencia de gestión de eventos.
- Reducción de errores operativos.
- Disponibilidad de reportes automatizados y datos centralizados.
- Aceptación y adopción del sistema por parte de usuarios finales.

10. Metodología de implementación

- Inicio: Definición de requisitos y diseño inicial.
- Desarrollo: Construcción de la API y pruebas unitarias.
- Implementación: Despliegue en contenedores Docker.
- Validación: Pruebas con usuarios finales y ajustes.
- Cierre: Documentación final y soporte técnico inicial.

11. Actores claves

- Equipo Técnico: Desarrolladores, arquitectos de software.
- Institución Educativa: Administradores, docentes y personal técnico.
- Usuarios Finales: Estudiantes y organizadores de eventos.
- Proveedores de Tecnología: Servicios en la nube, Docker Hub, MongoDB.

12. Papel y responsabilidades del personal

Gerente del Proyecto:

- Supervisar el desarrollo e implementación.
- Garantizar el cumplimiento de tiempos y presupuestos.

Desarrolladores:

- Implementar la API y realizar pruebas.
- Asegurar la integración continua.

Administradores del Sistema:

- Monitorear el funcionamiento y disponibilidad del sistema.

Usuarios Finales:

- Proporcionar retroalimentación para mejoras.

13. Plan de monitoreo y evaluación

Indicadores de Desempeño:

- Tiempo promedio de respuesta de la API.
- Número de eventos gestionados exitosamente.
- Porcentaje de adopción entre los usuarios.

Herramientas:

- Logs de monitoreo en la API.
- Reportes generados por la base de datos.

14. Cronograma del proyecto

Semana	Actividad
1	Levantamiento de requerimientos.
2-3	Diseño del modelo de datos en MongoDB.
4-5	Desarrollo de endpoints CRUD en .NET Core.
6-7	Configuración de Docker y pruebas iniciales.
8	Implementación de CI/CD con GitHub Actions.
9	Documentación de la API con Swagger UI.
10	Pruebas unitarias y validación final.
11	Despliegue y presentación del sistema.

15. Hitos de entregables

Semana 1: Levantamiento de Requerimientos

- Documentación de requisitos funcionales y no funcionales.
- Identificación de usuarios y necesidades del sistema.

Semana 2-3: Diseño del Modelo de Datos

- Modelo lógico y físico para MongoDB.
- Validación del diseño con los interesados.

Semana 4-5: Desarrollo de Endpoints CRUD

- Implementación de operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en .NET Core.
- Pruebas preliminares de funcionalidad de los endpoints.

Semana 6-7: Configuración de Docker y Pruebas Iniciales

- Creación del contenedor Docker para el despliegue de la API.
- Pruebas de compatibilidad y portabilidad entre entornos.

Semana 8: Implementación de CI/CD con GitHub Actions

- Configuración de flujos automatizados para integración y despliegue continuo.
- Validación del correcto funcionamiento de los pipelines.

Semana 9: Documentación de la API

- Creación de documentación interactiva con Swagger UI.
- Generación de guías de uso y ejemplos para desarrolladores.

Semana 10: Pruebas Unitarias y Validación Final

- Ejecución de pruebas unitarias para garantizar la calidad del código.
- Ajustes finales basados en resultados de validación.

Semana 11: Despliegue y Presentación del Sistema

- Despliegue de la API en un entorno productivo.
- Presentación del sistema a los usuarios finales e interesados.

II Presupuesto

1. Planteamiento de aplicación del presupuesto

- El presupuesto cubrirá desarrollo, pruebas, despliegue e infraestructura inicial para garantizar la sostenibilidad del sistema.

2. Presupuesto

Categoría	Costo Total (PEN)
Costos Generales	3,876.00
Costos Operativos	950.00
Costos del Ambiente	437.00
Costos de Personal	3,800.00
Total General	9,063.00

3. Análisis de Factibilidad

El proyecto es factible técnica, económica y operativamente, con una relación beneficio/costo estimada en 1.54.

4. Evaluación Financiera

Con beneficios proyectados de S/. 14,000 y costos de S/. 9,063, el proyecto presenta un retorno financiero positivo.

Anexo 01 – Requerimientos del Sistema

ID	Tipo de Requerimiento	Descripción	Prioridad
RF-01	Funcional	Permitir la creación de nuevos eventos a través de la API.	Alta
RF-02	Funcional	Consultar todos los eventos almacenados en la base de datos.	Alta
RF-03	Funcional	Consultar eventos por nombre mediante una búsqueda flexible.	Alta
RF-04	Funcional	Actualizar los datos de un evento existente.	Media
RF-05	Funcional	Eliminar un evento existente.	Media
RF-06	Funcional	Obtener los detalles de un evento específico por su ID.	Alta
RNF-01	No funcional - Rendimiento	La API debe responder a las solicitudes en menos de 500 ms en promedio.	Alta
RNF-02	No funcional - Escalabilidad	La base de datos debe soportar al menos 10,000 registros de eventos.	Media
RNF-03	No funcional - Despliegue	Desplegar la API en un contenedor Docker para facilitar el despliegue.	Alta
RNF-04	No funcional - Seguridad	Utilizar autenticación básica para proteger las rutas críticas de la API.	Media
RNF-05	No funcional - Mantenimiento	La API debe registrar los errores en un archivo o sistema de monitoreo.	Alta
RNF-06	No funcional - Pruebas	Implementar pruebas automatizadas para las rutas principales de la API.	Baja