UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

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

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

Sistema Proyecto Apis y Funciones Jarro_Valle
Documento de Arquitectura de Software

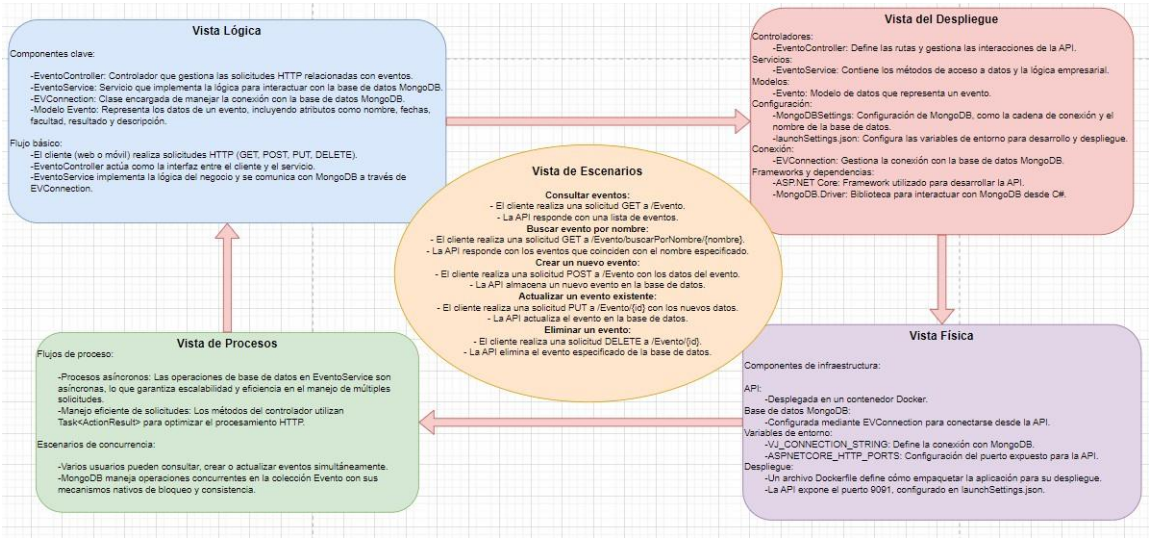
Versión *1.0*

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

INDICE GENERAL

1. INTRODUCCIÓN

1.1. Propósito (Diagrama 4+1)



1.2. Alcance

Describir la arquitectura de la API de eventos, desarrollada en C# .NET Framework y utilizando MongoDB como base de datos. El enfoque principal está en la vista lógica, abarcando los componentes fundamentales del sistema, su interacción y el modelo de datos.

1.3. Definición, siglas y abreviaturas

- API (Application Programming Interface): Conjunto de funciones y procedimientos que permite la interacción entre diferentes sistemas, en este caso entre la aplicación web/móvil y la API de eventos.
- CRUD: Acrónimo de Create, Read, Update, Delete; operaciones fundamentales de manipulación de datos en la base de datos.
- C#: Lenguaje de programación utilizado para desarrollar la API.
- Docker: Plataforma utilizada para desplegar la API en un contenedor.
- EVConnection: Clase encargada de gestionar la conexión con la base de datos MongoDB.
- Evento: Modelo de datos que representa un evento, incluyendo atributos como nombre, fechas, facultad, resultado y descripción.
- HTTP (Hypertext Transfer Protocol): Protocolo utilizado para las solicitudes entre clientes y la API.
- JSON (JavaScript Object Notation): Formato de intercambio de datos utilizado en las solicitudes y respuestas de la API.
- MongoDB: Base de datos NoSQL utilizada para almacenar la información de los eventos.
- SAD (Software Architecture Design): Documento que describe la arquitectura del software y su diseño.
- Swagger: Herramienta integrada para documentar y probar las rutas y funcionalidades de la API.

1.4. Organización del proyecto

- FD01-EPIS-Informe de Factibilidad.docx
- FD02-EPIS-Informe Visión.docx
- FD03-EPIS-Informe Especificación Requerimientos.docx
- FD04-EPIS-Informe Arquitectura de Software.docx
- FD05-EPIS-Informe ProyectoFinal.docx
- FD06-EPIS-PropuestaProyecto.docx

2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS

2.1. Priorización de requerimientos

2.1.1. Requerimientos Funcionales

ID	*Descripción*	*Prioridad*
RF-01	Permitir la creación de nuevos eventos a través de la API.	Alta
RF-02	Consultar todos los eventos almacenados en la base de datos.	Alta
RF-03	Consultar eventos por nombre mediante una búsqueda flexible.	Alta
RF-04	Actualizar los datos de un evento existente.	Media
RF-05	Eliminar un evento existente.	Media
RF-06	Obtener los detalles de un evento específico por su ID.	Alta

2.1.2. Requerimientos No Funcionales – Atributos de Calidad

ID	*Descripción*	*Prioridad*
RNF-01	La API debe responder a las solicitudes en menos de 500 ms en promedio.	Alta
RNF-02	La base de datos debe soportar al menos 10,000 registros de eventos.	Media
RNF-03	Desplegar la API en un contenedor Docker para facilitar el despliegue.	Alta
RNF-04	Utilizar autenticación básica para proteger las rutas críticas de la API.	Media
RNF-05	La API debe registrar los errores en un archivo o sistema de monitoreo.	Alta
RNF-06	Implementar pruebas automatizadas para las rutas principales de la API.	Baja

2.2. Restricciones

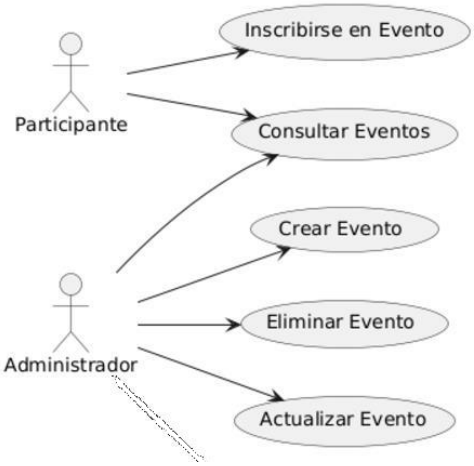
- Framework Obligatorio: La API debe desarrollarse utilizando C# .NET Framework y no se permite el uso de otros frameworks o lenguajes de programación.
- Base de Datos: La solución debe utilizar MongoDB como base de datos principal, sin posibilidad de integrar otros sistemas de almacenamiento.

- Despliegue Contenerizado: El sistema debe ser desplegado obligatoriamente en un contenedor Docker.
- Entorno de Ejecución: El entorno de producción debe configurarse usando variables de entorno, siguiendo estándares de seguridad para las credenciales y conexiones.
- Interfaz de Usuario: No se incluye el desarrollo de interfaces gráficas; la API se limitará a responder a solicitudes HTTP.

3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

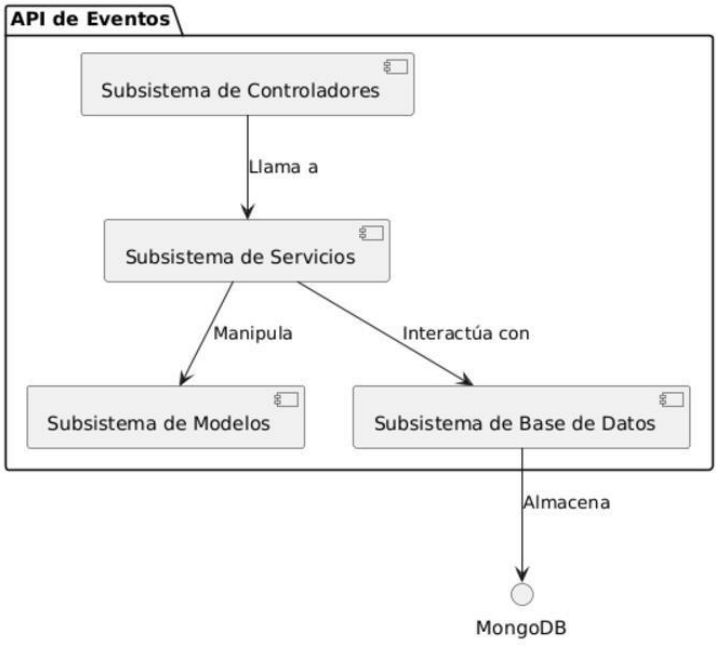
3.1. Vista de Caso de uso

3.1.1. Diagramas de Casos de uso



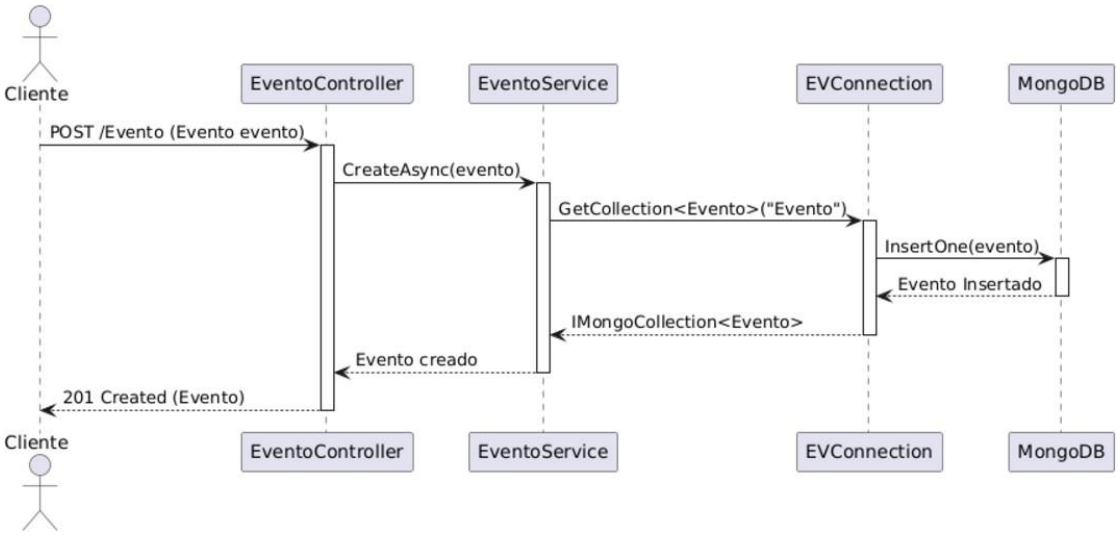
3.2. Vista Lógica

3.2.1. Diagrama de Subsistemas (paquetes)

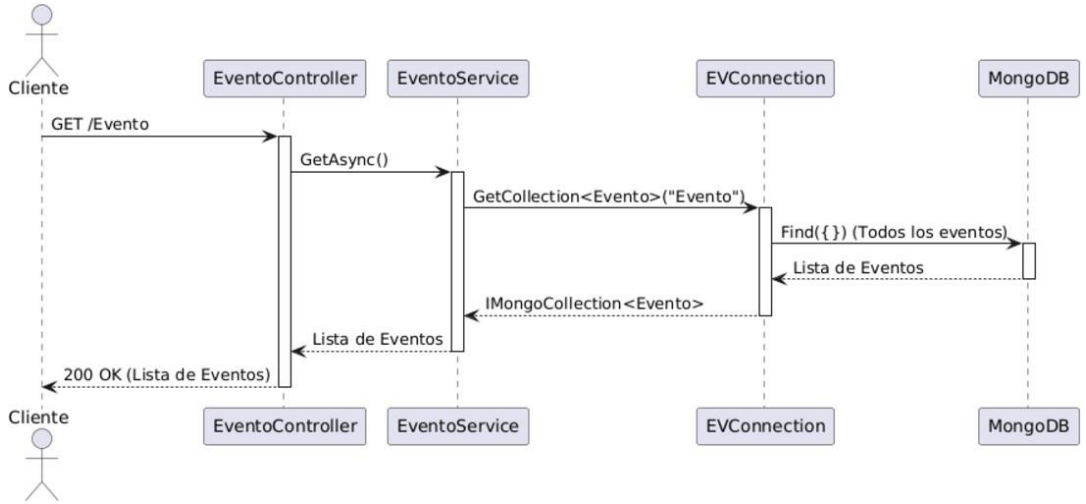


3.2.2. Diagrama de Secuencia

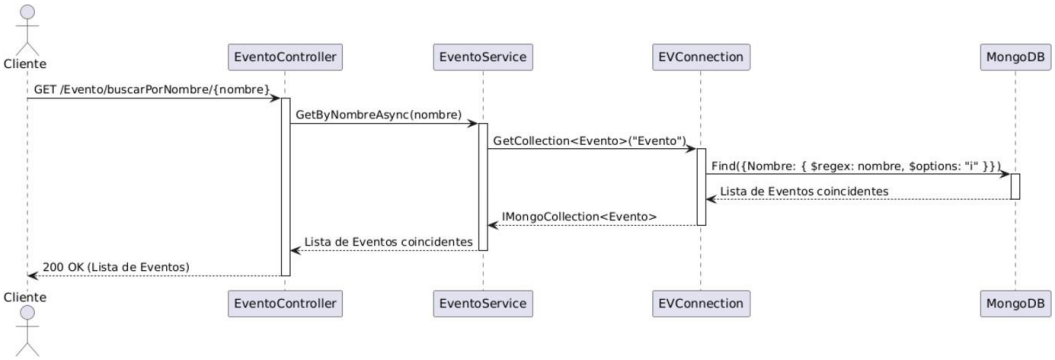
- RF-01 Permitir la creación de nuevos eventos a través de la API



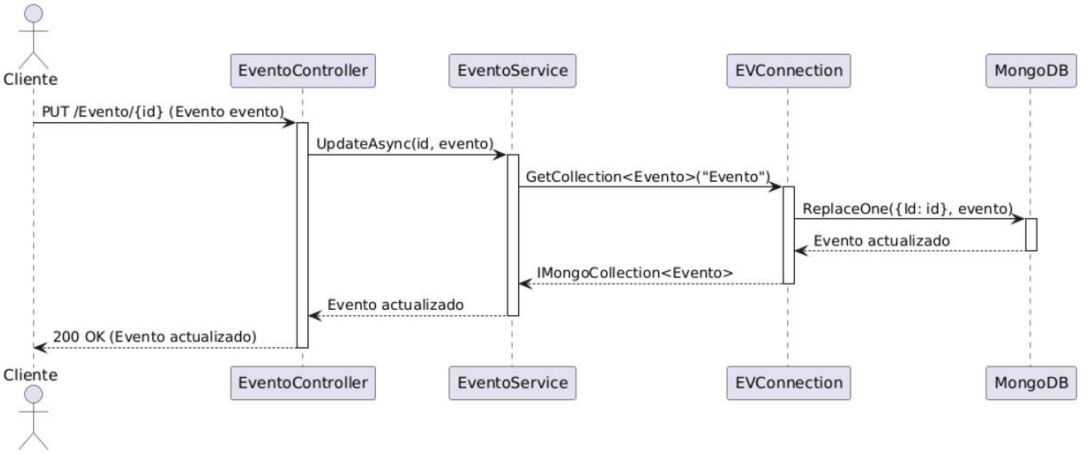
- RF-02 Consultar todos los eventos almacenados en la base de datos.



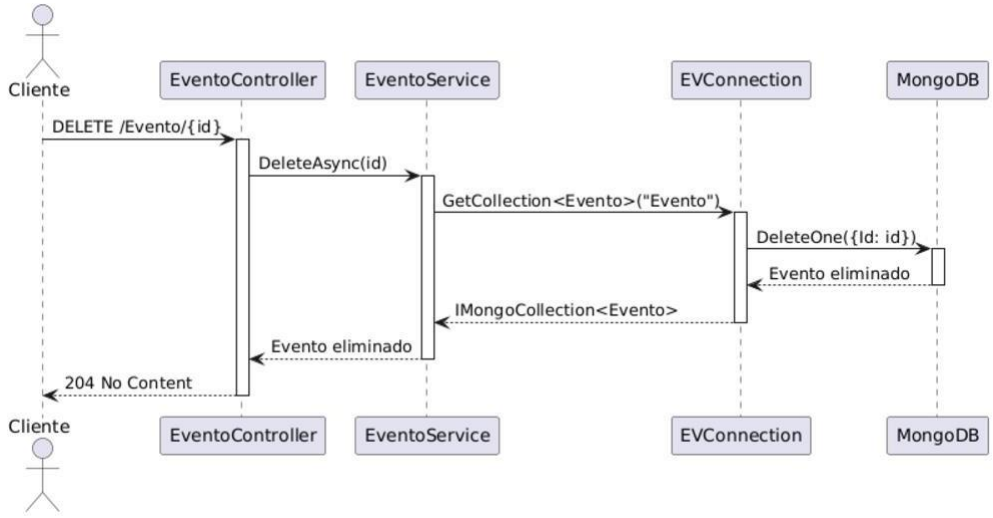
- RF-03 Consultar eventos por nombre mediante una búsqueda flexible.



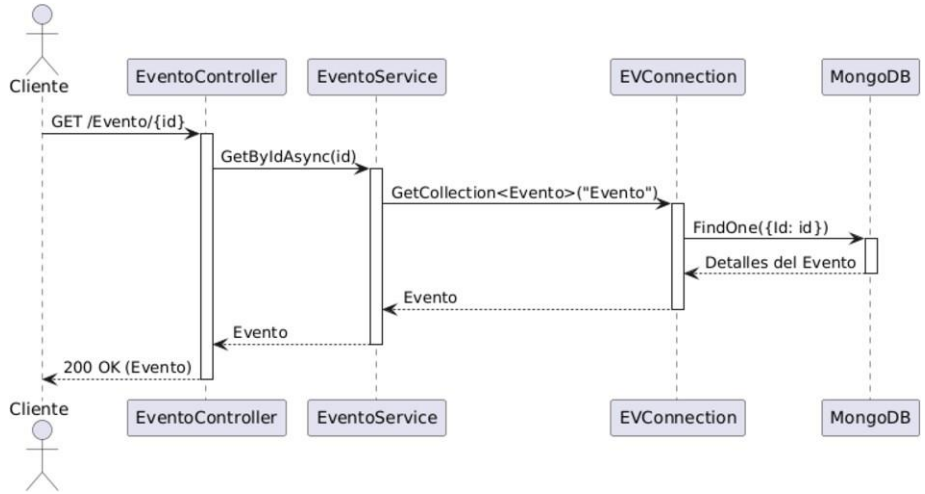
- RF-04 Actualizar los datos de un evento existente.



- RF-05 Eliminar un evento existente.



- RF-06 Obtener los detalles de un evento específico por su ID.



3.2.3. Diagrama de Colaboración (vista de diseño)

3.2.4. Diagrama de Objetos

- RF-01 Permitir la creación de nuevos eventos a través de la API

- RF-02 Consultar todos los eventos almacenados en la base de datos.

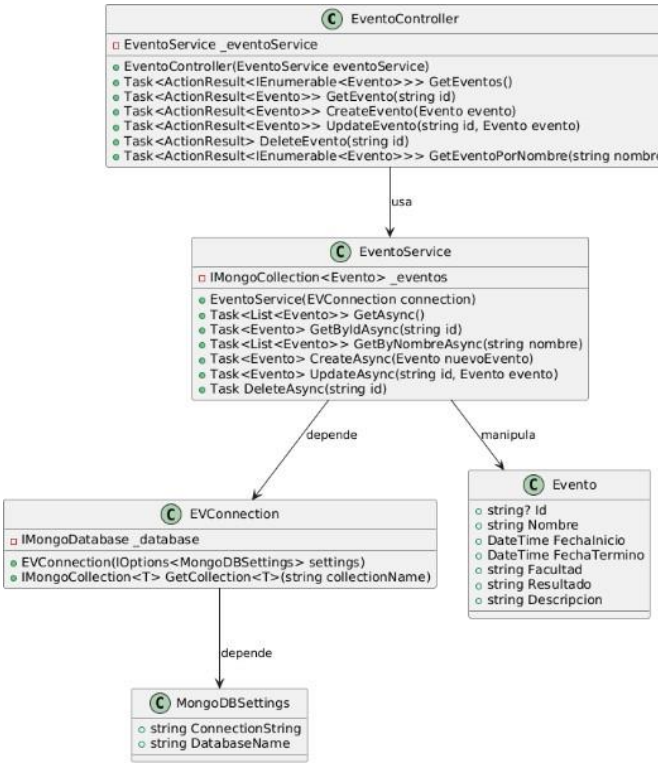
- RF-03 Consultar eventos por nombre mediante una búsqueda flexible.

- RF-04 Actualizar los datos de un evento existente.

- RF-05 Eliminar un evento existente.

- RF-06 Obtener los detalles de un evento específico por su ID.

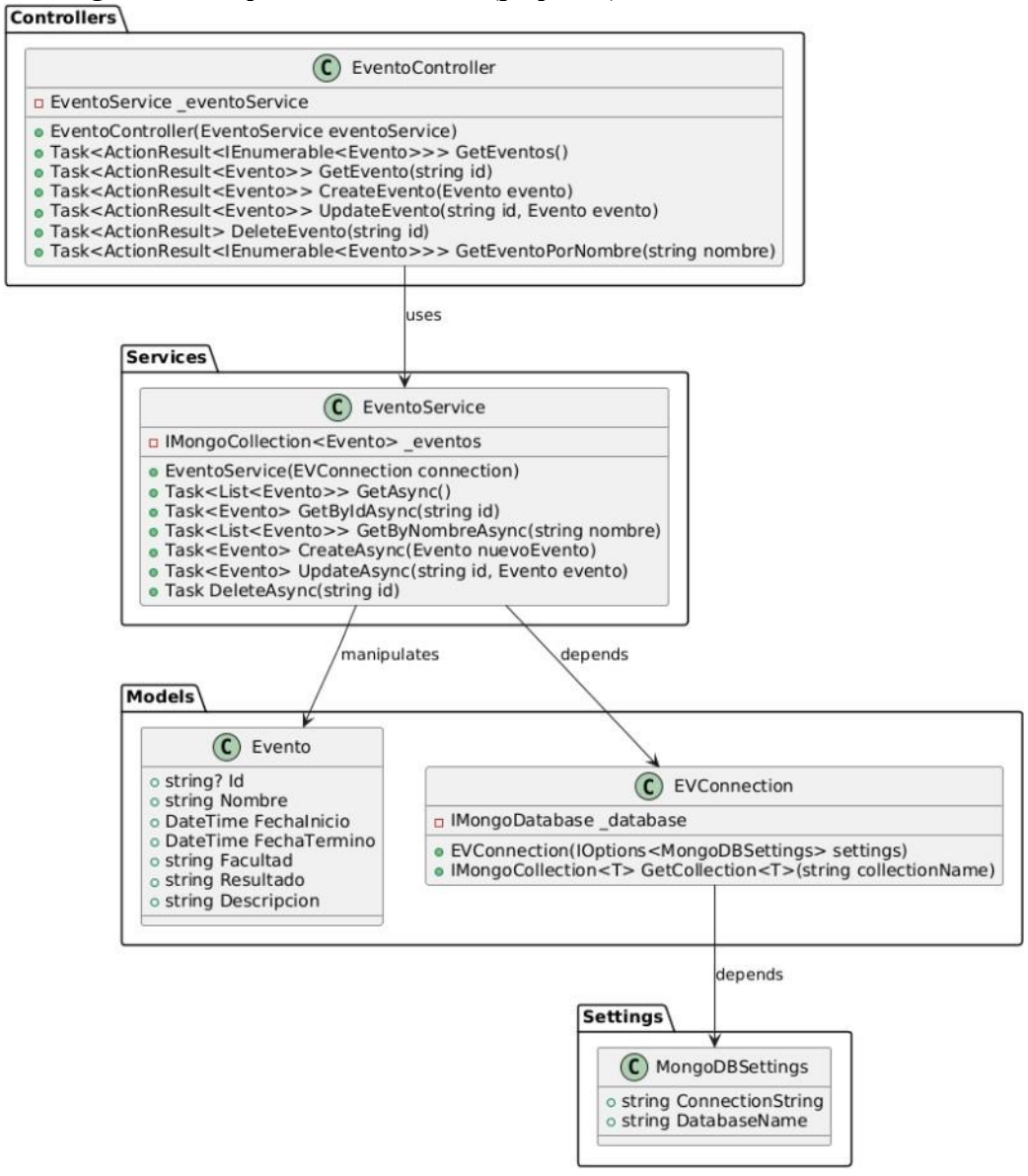
3.2.5. Diagrama de Clases



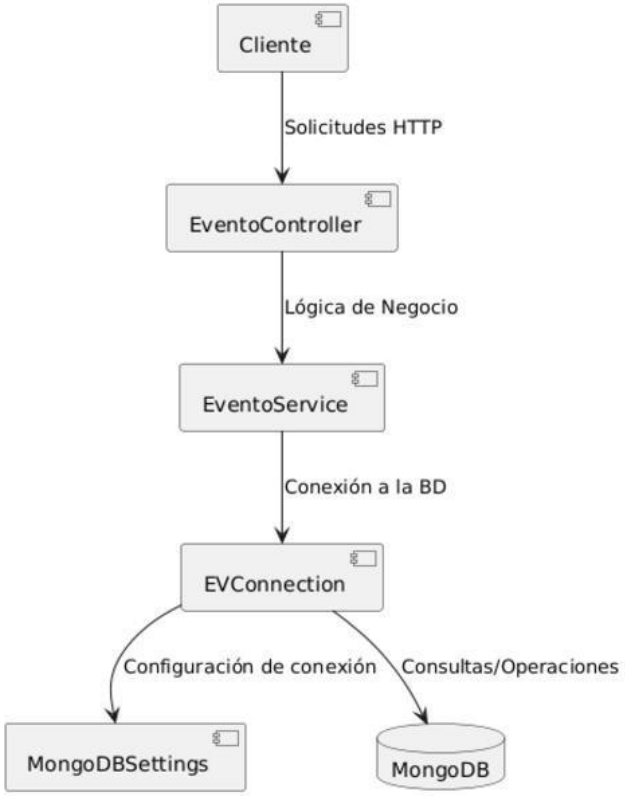
3.2.6. Diagrama de Base de datos

3.3. Vista de Implementación (vista de desarrollo)

3.3.1. Diagrama de arquitectura software (paquetes)

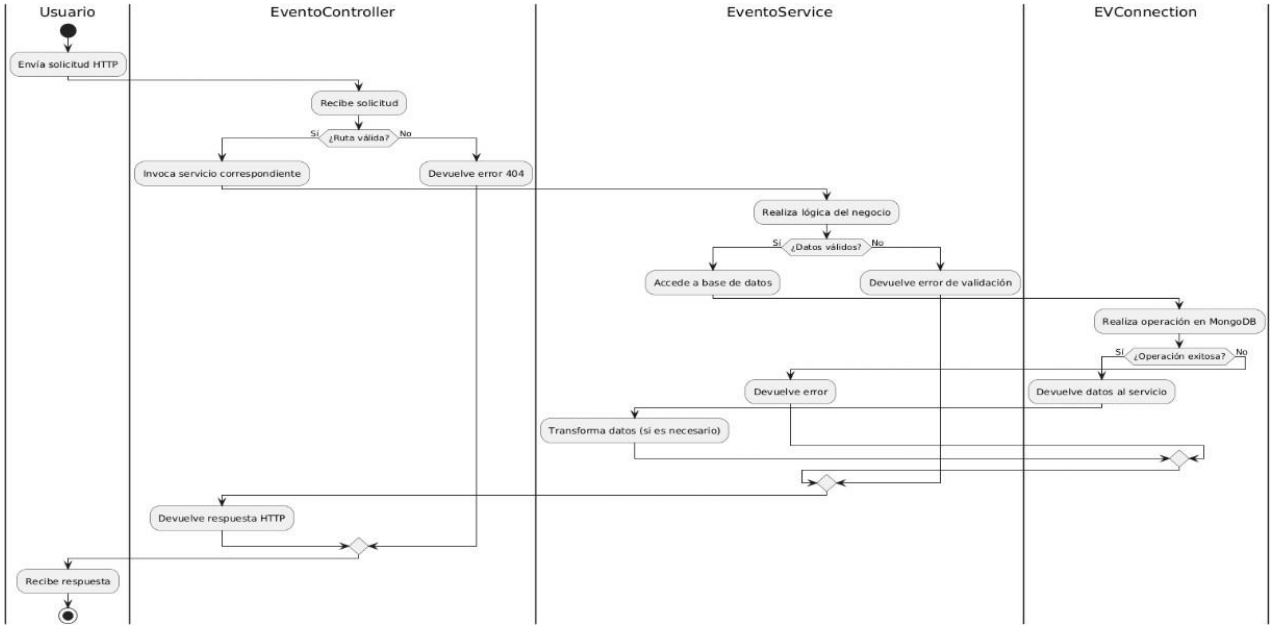


3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)



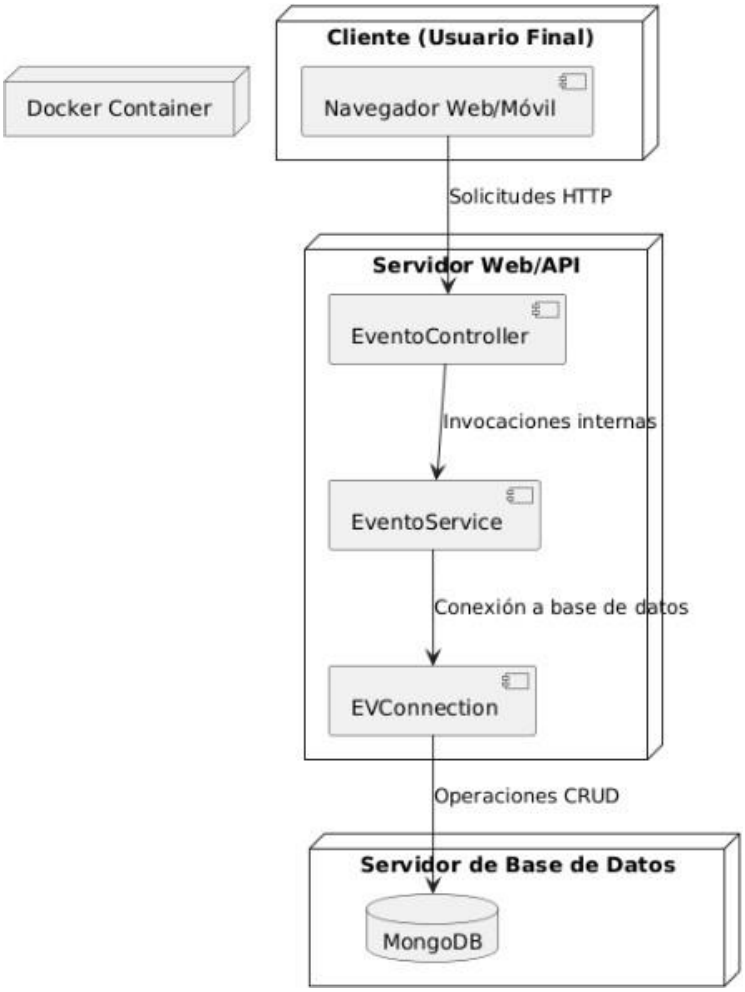
3.4. Vista de procesos

3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)



3.5. Vista de Despliegue (vista física)

3.5.1. Diagrama de despliegue



4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

Escenario de Funcionalidad

La API de eventos permite realizar operaciones CRUD (crear, leer, actualizar, eliminar) sobre los eventos almacenados en la base de datos. Además, ofrece funcionalidades específicas como la búsqueda de eventos por nombre y la consulta detallada de eventos a través de su ID. Estas funcionalidades están diseñadas para cumplir con los requerimientos funcionales, garantizando también la seguridad de las solicitudes mediante la implementación de autenticación básica.

Escenario de Usabilidad

La API está diseñada para que los desarrolladores puedan integrarla fácilmente con aplicaciones cliente, tanto móviles como web. Cuenta con documentación clara generada automáticamente (Swagger), lo que facilita el aprendizaje y uso de sus endpoints. Además, utiliza formatos de datos estándar como JSON, que son ampliamente conocidos y manejables por los desarrolladores, optimizando la interpretación y manipulación de datos.