

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto A*PI y Funciones de Lugares***

Curso: *Tópicos de Base de Datos Avanzados*

Docente: *Patrick Jose Cuadros Quiroga*

Alumno:

***Mayner Gonzalo Anahua Coaquira (2020067145)***

**Tacna – Perú**

***2024***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

Sistema *API y Funciones de Lugares*

Documento de Visión

Versión *1.0*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

**INDICE GENERAL**

1. Introducción 1

1.1 Propósito 1

1.2 Alcance 1

1.3 Definiciones, Siglas y Abreviaturas 1

1.4 Referencias 1

1.5 Visión General 1

2. Posicionamiento 1

2.1 Oportunidad de negocio 1

2.2 Definición del problema 2

3. Descripción de los interesados y usuarios 3

3.1 Resumen de los interesados 3

3.2 Resumen de los usuarios 3

3.3 Entorno de usuario 4

3.4 Perfiles de los interesados 4

3.5 Perfiles de los Usuarios 4

3.6 Necesidades de los interesados y usuarios 6

4. Vista General del Producto 7

4.1 Perspectiva del producto 7

4.2 Resumen de capacidades 8

4.3 Suposiciones y dependencias 8

4.4 Costos y precios 9

4.5 Licenciamiento e instalación 9

5. Características del producto 9

6. Restricciones 10

7. Rangos de calidad 10

8. Precedencia y Prioridad 10

9. Otros requerimientos del producto 10

[b) Estandares legales](#_Toc394513800) 32

[c) Estandares de comunicación](#_Toc394513800) 37

[d) Estandaraes de cumplimiento de la plataforma](#_Toc394513800) 42

[e) Estandaraes de calidad y seguridad](#_Toc394513800) 42

[CONCLUSIONES](#_Toc394513803) 46

[RECOMENDACIONES](#_Toc394513804) 46

[BIBLIOGRAFIA](#_Toc394513805) 46

[WEBGRAFIA](#_Toc394513806) 46

**1. Introducción**

**1.1 Propósito**

El propósito de este informe es presentar la visión general del proyecto API de Lugares, un sistema desarrollado para gestionar lugares, direcciones y categorías de una manera eficiente y escalable. Este informe tiene como objetivo definir los aspectos clave del proyecto, incluyendo los interesados, los usuarios, los requisitos, y la arquitectura del sistema.

**1.2 Alcance**

Este proyecto incluye el desarrollo de una API RESTful que permitirá a los usuarios realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre lugares, direcciones y categorías. La API está diseñada para ser implementada utilizando FastAPI como framework y CouchDB como base de datos, con despliegue simplificado mediante Docker Compose. El proyecto se desplegará y se pondrá a disposición de los usuarios finales a través de una interfaz que permita la integración con otros sistemas.

**1.3 Definiciones, Siglas y Abreviaturas**

* API: Interfaz de Programación de Aplicaciones.
* FastAPI: Framework web para la construcción de APIs rápidas y eficientes.
* CouchDB: Base de datos NoSQL orientada a documentos.
* Docker: Plataforma para crear, desplegar y ejecutar aplicaciones en contenedores.

**1.4 Referencias**

* FastAPI: <https://fastapi.tiangolo.com/>
* CouchDB: <https://couchdb.apache.org/>
* Docker: <https://docs.docker.com/>

**1.5 Visión General**

Este informe proporciona un resumen completo del proyecto API de Lugares, detallando los requisitos, las dependencias, la arquitectura, los costos de implementación y los beneficios del sistema. El proyecto es esencial para mejorar la gestión de la información de lugares, direcciones y categorías, permitiendo su integración en aplicaciones de terceros, y se desarrollará con un enfoque en la escalabilidad y seguridad.

**2. Posicionamiento**

**2.1 Oportunidad de negocio**

La implementación de la API de Lugares ofrece una solución que mejora la eficiencia y accesibilidad de la gestión de datos relacionados con lugares y categorías. Al utilizar FastAPI y CouchDB, la API proporcionará una plataforma escalable y rápida para manejar grandes volúmenes de datos, siendo de gran utilidad para las empresas que necesiten integrar esta información en sus aplicaciones. Además, su disponibilidad mediante Docker facilita el despliegue, lo que permite reducir costos operativos y mejorar el tiempo de respuesta.

**2.2 Definición del problema**

La gestión de lugares y sus respectivas categorías de manera manual o en sistemas poco integrados puede generar ineficiencias y falta de accesibilidad. Este proyecto busca resolver esos problemas proporcionando una plataforma centralizada y eficiente para manejar los datos, accesible mediante una API RESTful y lista para ser integrada con otros sistemas.

**3. Descripción de los interesados y usuarios**

**3.1 Resumen de los interesados**

Los interesados principales del proyecto son:

* Desarrolladores que implementarán la API y sus respectivas integraciones.
* Usuarios finales que harán uso de la API para gestionar lugares, direcciones y categorías.
* Administrador de la infraestructura que manejará los entornos de desarrollo, pruebas y producción.

**3.2 Resumen de los usuarios**

Los usuarios finales incluyen a las empresas o individuos que necesiten gestionar información de lugares y categorías, y a las aplicaciones que integrarán esta API para acceder a estos datos.

**3.3 Entorno de usuario**

El entorno de usuario incluirá acceso tanto para administradores como para aplicaciones que consumirán la API. Los administradores necesitarán autenticación y autorización para realizar cambios en los datos, mientras que las aplicaciones externas consumirán la API de manera solo lectura o mediante acceso limitado según los permisos definidos.

**3.4 Perfiles de los interesados**

* Desarrolladores: Serán responsables del diseño e implementación de la API y la infraestructura de bases de datos.
* Usuarios finales: Necesitarán una interfaz para interactuar con la API y gestionar la información.

**3.5 Perfiles de los Usuarios**

* Administrador de la API: Persona encargada de configurar y gestionar el acceso a la API.
* Desarrolladores de aplicaciones: Consumirán la API desde sus aplicaciones externas, como sitios web o apps móviles.

**3.6 Necesidades de los interesados y usuarios**

Los usuarios requieren que la API sea escalable, segura y eficiente. Los interesados necesitan una solución que pueda ser fácilmente desplegada y gestionada, con una arquitectura flexible que permita futuras integraciones y ampliaciones.

**4. Vista General del Producto**

**4.1 Perspectiva del producto**

La API de Lugares estará compuesta por una serie de endpoints RESTful que permitirán realizar operaciones CRUD en los datos de lugares, direcciones y categorías. La API será accesible a través de solicitudes HTTP, y la base de datos estará implementada utilizando CouchDB para permitir una gestión eficiente de los datos. Docker se utilizará para simplificar el despliegue.

**4.2 Resumen de capacidades**

La API proporcionará las siguientes capacidades:

* Gestión de lugares, direcciones y categorías mediante CRUD.
* Escalabilidad y seguridad en el manejo de grandes volúmenes de datos.
* Despliegue simplificado mediante Docker Compose.
* Integración con aplicaciones externas.

**4.3 Suposiciones y dependencias**

El proyecto depende de la disponibilidad de CouchDB como base de datos y de FastAPI como framework. También se asume que el entorno de ejecución (Docker) estará correctamente configurado para soportar el despliegue de la API.

**4.4 Costos y precios**

Los costos asociados al desarrollo del proyecto se dividen en:

|  |  |
| --- | --- |
| Concepto | Costo Estimado |
| Red y conectividad | $300 |
| Servidores en la nube | $1,000 |

**4.5 Licenciamiento e instalación**

El sistema será de código abierto y se distribuirá bajo una licencia MIT. La instalación será sencilla mediante Docker Compose, lo que permitirá a los usuarios implementar el sistema rápidamente.

**5. Características del producto**

El producto contará con las siguientes características:

* API RESTful que permita gestionar información de lugares, direcciones y categorías.
* Despliegue a través de Docker.

**6. Restricciones**

El sistema no incluirá funcionalidades de procesamiento de pagos o gestión directa de reservas, ya que se centrará exclusivamente en la gestión de información.

**7. Rangos de calidad**

El sistema debe tener un rendimiento que permita responder a las solicitudes en menos de 200ms y debe estar disponible 90.0% del tiempo.

**8. Precedencia y Prioridad**

La prioridad será asegurar la estabilidad de la API y su capacidad para manejar grandes volúmenes de datos de manera eficiente. La integración con aplicaciones móviles y web externas será secundaria.

**9. Otros requerimientos del producto**

**b) Estándares legales**

El sistema debe cumplir con las leyes locales sobre protección de datos y privacidad de la información.

**c) Estándares de comunicación**

La API debe usar HTTPS para todas las comunicaciones, asegurando la protección de los datos durante su transmisión.

**d) Estándares de cumplimiento de la plataforma**

El sistema debe ser compatible con navegadores web modernos y dispositivos móviles.

**e) Estándares de calidad y seguridad**

El sistema debe implementar autenticación y autorización para acceder a los endpoints y cifrado para la transmisión de datos sensibles.

**CONCLUSIONES**

El proyecto API de Lugares presenta una solución robusta y escalable para la gestión de lugares, direcciones y categorías. La combinación de FastAPI, CouchDB y Docker asegura una implementación eficiente y flexible, que puede adaptarse a futuras necesidades.

**RECOMENDACIONES**

Se recomienda proceder con el desarrollo de la API según lo planificado, garantizando que los requisitos de calidad y seguridad sean cumplidos a lo largo de todas las fases del proyecto.

**BIBLIOGRAFÍA**

* <https://fastapi.tiangolo.com/>
* <https://couchdb.apache.org/>

**WEBGRAFÍA**

* <https://docs.docker.com/>