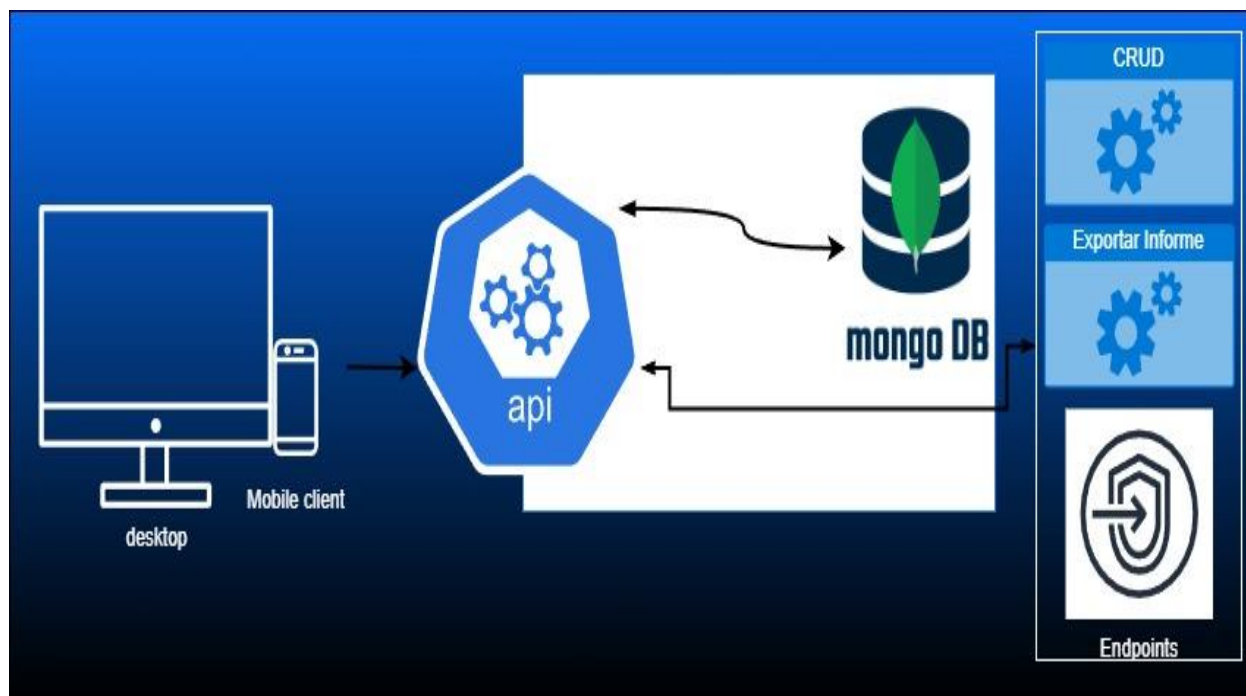


## Sistema de Gestión de Materiales para Obras de Construcción

- Enlace del consumo del Api: <https://consumoapi.fly.dev/>
- Enlace código fuente de la Api RestFull:  
<https://github.com/Cristian19C/Taller2ApiRestFull>
- Enlace código fuente del Consumo de la Api  
<https://github.com/Cristian19C/Taller2ConsumoApiRestFull>

### 1. Modelo de la arquitectura



*Imagen 1. Modelo*

El modelo de arquitectura de servicios, representado en la **Imagen 1**, describe la operación de una API RESTFULL que tiene como objetivo facilitar la gestión del inventario de materiales en obras de construcción o civiles. Los usuarios acceden a la API a través de dispositivos y realizan solicitudes de servicios, como visualizar la lista de materiales. Estos servicios interactúan con una base de datos MongoDB alojada en MongoDB Atlas, que actúa como el almacén principal de datos para la administración del inventario.

La API ofrece una interfaz de programación de aplicaciones (API) que define diversos EndPoints que corresponden a las operaciones CRUD (Crear, Leer, Actualizar, Borrar) para el manejo de materiales en el inventario. Cada Endpoint permite a los usuarios realizar acciones específicas, como agregar nuevos materiales, obtener detalles de los materiales existentes, actualizar información y eliminar registros de materiales.

Para que los usuarios puedan acceder a través de protocolos HTTP a los servicios mencionados se gestiona una aplicación en Fly.io, la cual se asigna un subdominio bajo el dominio de “Fly.io”, haciendo posible acceder a dichos servicios a través de internet de forma fácil.

Cuando los usuarios realizan solicitudes a través de la API, los datos se procesan mediante las operaciones correspondientes y se realiza la interacción con la base de datos MongoDB para realizar las acciones solicitadas. Por ejemplo, cuando un usuario desea ver la lista de materiales, la API consulta la base de datos para obtener los datos requeridos y devuelve la información en formato JSON al cliente.

La API también proporciona capacidades de búsqueda y filtrado para que los usuarios puedan encontrar rápidamente los materiales deseados en el inventario. Además, la seguridad y la autorización son aspectos esenciales, por lo que la API implementa mecanismos para garantizar que solo los usuarios autorizados puedan acceder y manipular la información del inventario. Además de las operaciones CRUD, la API permite la generación de informes PDF relacionados con el inventario de materiales, brindando a los usuarios una visión general de los niveles de stock.

## **2. Documentación API Sistema de Gestión de Materiales para Obras de Construcción**

Descripción de los **Endpoints** que permiten la realización de las operaciones del **CRUD**, sobre los registros de los materiales, haciendo unos de los diferentes métodos del protocolo **HTTP**

**URL:** <https://consumoapi.fly.dev/inventory>

### **2.1 Obtener todos los registros de materiales del inventario**

- **Método HTTP:** GET
- **Ruta:** ‘/’
- **Descripción:** Obtiene todos los materiales que están almacenados en la base de datos en la colección inventory
- **Respuesta exitosa (200):**

```
{  
  "status": true,  
  "data": [Array de objetos materiales]  
}
```

- **Respuesta de error (500):**

```
{  
  "status": false,  
  "data": [Mensaje de error]  
}
```

## 2.2 Obtener un material por su id

- **Método HTTP:** GET
- **Ruta:** ‘/:id’
- **Descripción:** Obtiene un material especificado
- **Respuesta exitosa (200):**

```
{  
  "status": true,  
  "data": [Objeto del material]  
}
```

- **Respuesta de error (500):**

```
{  
  "status": false,  
  "data": [Mensaje de error]  
}
```

## 2.3 Agregar un nuevo material al inventario

- **Método HTTP:** POST
- **Ruta:** ‘/’
- **Descripción:** Agregar un nuevo material a la base de datos
- **Solicitud:** Objeto en formato JSON con sus propiedades

```
{
  id_product: id material,
  name: Nombre,
  price: Precio,
  number: Cantidad,
  description: description,
  brand: brand
}
```

- **Respuesta exitosa (200):**

```
{
  "status": true,
  "data": [Objeto del material creado]
}
```

- **Respuesta de error (500):**

```
{
  "status": false,
  "data": [Mensaje de error]
}
```

## 2.4 Actualizar un material por su id del material

- **Método HTTP:** PUT
- **Ruta:** '/:id'
- **Descripción:** Actualiza un material específico a través del id del material
- **Solicitud:** Objeto en formato JSON con sus propiedades

```
{
  id_product: id material,
  name: Nombre,
  price: Precio,
  number: Cantidad,
  description: description,
  brand: brand
}
```

- **Respuesta exitosa (200):**  

```
{
  "status": true,
  "data": [Objeto del material actualizado ]
}
```
- **Respuesta de error (500):**  

```
{
  "status": false,
  "data": [Mensaje de error]
}
```

## 2.5 Eliminar un vehículo por id del material

- **Método HTTP:** DELETE
- **Ruta:** '/:id'
- **Descripción:** Elimina un material específico a través del id del material
- **Respuesta exitosa (200):**  

```
{
  "status": true,
  "data": [Objeto del material eliminado]
}
```
- **Respuesta de error (500):**  

```
{
  "status": false,
  "data": [Mensaje de error]
}
```