

Consideraciones generales:	2
Investigar una API	3
Recursos clave	3
Funciones principales:	3
Buscar publicaciones	3
Los endpoints de una API	3
esearch.fcgi	3
Parámetros importantes:	3
efetch.fcgi	3
Parámetros importantes:	3
esummary.fcgi	4
Parámetros importantes:	4
elink.fcgi	4
Parámetros importantes:	4
Requisitos de uso	4
Tasa de solicitudes	4
Documentación sobre la clave de API	4
Formatos de respuesta	4
Autenticación	4
Implementar una solución para consumir la API	5
Diseñar un esquema para almacenar los datos (MongoDB)	5
Repositorio de GitHub	5
Despliegue CI/CD	6
CI (Integración Continua)	6
Beneficios del CI:	6
CD (Entrega y Despliegue Continuo)	6
Entrega Continua (Continuous Delivery):	6
Despliegue Continuo (Continuous Deployment):	6
Beneficios del CD:	6
Implementación	7

Consideraciones generales:

- **GitHub:**
 - Cada API debe tener su propio repositorio o un repositorio central con carpetas separadas para cada solución.
 - Documentar el código y las instrucciones de uso en un archivo `README.md`.
- **Base de datos:**
 - Usar una base de datos que soporte el almacenamiento y consulta eficiente de los datos (MongoDB).
 - Proveer scripts o configuraciones para inicializar la base de datos.
- **Despliegue:**
 - Utilizar un entorno de despliegue (como AWS, Heroku, Render, o Github Actions) para mostrar el funcionamiento de cada API.
 - Documentar los pasos de despliegue en el repositorio correspondiente.

Investigar una API

La API de PubMed, conocida como Entrez Programming Utilities (E-utilities), permite acceder a datos de publicaciones científicas.

Recursos clave

Documentación oficial: E-utilities

Endpoint principal: <https://eutils.ncbi.nlm.nih.gov/entrez/eutils/>

Funciones principales:

esearch: Buscar publicaciones.

efetch: Obtener detalles completos de las publicaciones.

esummary: Resumen de las publicaciones.

Buscar publicaciones

Los endpoints de una API

Son las direcciones o URLs específicas que proporciona una API para interactuar con ella y realizar diferentes operaciones. Cada endpoint corresponde a una funcionalidad específica de la API, como buscar, crear, actualizar o eliminar datos.

esearch.fcgi

Busca publicaciones y devuelve los identificadores únicos (PubMed IDs, PMIDs).

URL: <https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi>

Parámetros importantes:

db: Base de datos, en este caso "pubmed".

term: Término de búsqueda (palabras clave).

retmax: Número máximo de resultados a devolver (por defecto, 20).

retmode: Formato de respuesta (xml o json).

Ejemplo:

<https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term=machine+learning&retmax=10&retmode=json>

efetch.fcgi

Recupera detalles completos de una publicación usando su PMID.

URL: <https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi>

Parámetros importantes:

db: Base de datos, "pubmed".

id: Lista de PMIDs separados por comas.

retmode: Formato de respuesta (xml o text).

Ejemplo:

<https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=pubmed&id=123456,7891011&retmode=xml>

esummary.fcgi

Obtiene un resumen breve de publicaciones basado en PMIDs.

URL: <https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esummary.fcgi>

Parámetros importantes:

db: Base de datos, "pubmed".

id: Lista de PMIDs separados por comas.

retmode: Formato de respuesta (xml o json).

elink.fcgi

Encuentra enlaces entre publicaciones y otras bases de datos.

URL: <https://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi>

Parámetros importantes:

dbfrom: Base de datos de origen (por ejemplo, "pubmed").

id: PMID o lista de PMIDs.

linkname: Tipo de enlace (por ejemplo, referencias citadas).

Requisitos de uso

Tasa de solicitudes

La API permite hasta 3 solicitudes por segundo. Si necesitas más, puedes registrar una clave de API gratuita.

Documentación sobre la clave de API

Key Registration

Formatos de respuesta

Puedes obtener respuestas en JSON, XML o texto.

Autenticación

Aunque básica, la clave de API mejora la capacidad de solicitudes (hasta 10 por segundo).

**Implementar una solución para
consumir la API**

**Diseñar un esquema para
almacenar los datos
(MongoDB)**

Repositorio de GitHub

Despliegue CI/CD

Un despliegue CI/CD (Integración Continua/Despliegue Continuo, por sus siglas en inglés) es una práctica en desarrollo de software que automatiza el proceso de construcción, prueba y despliegue de aplicaciones. Es clave en entornos modernos de desarrollo, ya que asegura que los cambios en el código puedan integrarse y enviarse a producción de forma rápida, confiable y con menos errores humanos.

CI (Integración Continua)

La Integración Continua se refiere al proceso automatizado en el que los cambios en el código (por ejemplo, un commit en Git) son:

1. Construidos.
2. Probados automáticamente.
3. Verificados para asegurar que no rompan la aplicación.

Esto permite a los equipos trabajar en paralelo sin preocuparse demasiado por conflictos o errores, porque cualquier problema será detectado en las etapas tempranas de desarrollo.

Beneficios del CI:

- Detectar errores rápidamente.
- Evitar problemas al integrar diferentes partes del código.
- Asegurarse de que el software siempre esté en un estado funcional.

CD (Entrega y Despliegue Continuo)

El CD tiene dos variantes:

Entrega Continua (Continuous Delivery):

Los cambios que pasan las pruebas automatizadas se colocan en un entorno listo para ser desplegado en producción, pero aún requieren aprobación manual para ir a producción.

Despliegue Continuo (Continuous Deployment):

Los cambios pasan automáticamente a producción sin intervención humana. Este proceso requiere una alta confianza en las pruebas automatizadas.

Beneficios del CD:

- Despliegues más rápidos y frecuentes.

- Reducir el tiempo necesario para llevar nuevas funcionalidades a los usuarios.
- Minimizar riesgos al hacer despliegues pequeños y frecuentes en lugar de grandes lanzamientos ocasionales.

Implementación

Para implementar CI/CD, puedes usar GitHub Actions. Crea un archivo en `.github/workflows/ci.yml`: