

## DESCRIPCIÓN DEL ANTEPROYECTO

### TÍTULO: NistDataConverter – Conversor Web de Datos del NIST a JSON y CSV

#### Introducción

Durante el periodo de prácticas del ciclo formativo DAW, se presentó la necesidad de automatizar la consulta de datos ofrecidos por el NIST (National Institute of Standards and Technology), una fuente de referencia a nivel mundial en materia de estándares, ciberseguridad y datos técnicos. La solución propuesta entonces fue una herramienta sencilla en Python, ejecutada desde consola, que permitía acceder a cierta información puntual. Sin embargo, esta solución, aunque funcional, carecía de usabilidad y no estaba pensada para usuarios sin conocimientos técnicos. A partir de esa experiencia, surgió la idea de crear una versión mucho más completa, accesible y profesional: una aplicación web con interfaz gráfica que permitiera consultar, filtrar y exportar los datos del NIST de forma flexible y eficiente.

**NistDataConverter** nace con el objetivo de cubrir esa carencia detectada en la práctica real, y al mismo tiempo ser una herramienta útil para estudiantes, investigadores y técnicos que necesiten trabajar con datos estandarizados. La aplicación permitirá aplicar filtros como el rango de fechas o el nivel de severidad (por ejemplo, en registros de vulnerabilidades), y exportar los resultados en formatos reutilizables como JSON y CSV. Con este enfoque, se busca no solo mejorar la utilidad creada en el entorno de prácticas, sino también crear un producto adaptable a distintos entornos académicos o profesionales donde el acceso ágil a datos técnicos sea una necesidad.

#### Objetivos

El objetivo de este proyecto es proporcionar una aplicación web denominada **NistDataConverter**, diseñada para amoldarse a las necesidades de estudiantes, investigadores y técnicos que requieren un acceso ágil y eficiente a los datos publicados por el NIST (National Institute of Standards and Technology). Esta herramienta facilitará la consulta, el filtrado personalizado y la exportación de información relevante.

Para alcanzar dicho objetivo general, la aplicación contará con las siguientes funcionalidades principales, estructuradas para mejorar la usabilidad y el acceso a la información:

- **Conexión con la API del NIST:** Se establecerá una conexión directa y en tiempo real con la API pública del NIST para la obtención de datos actualizados.
- **Interfaz Web Intuitiva:** Se diseñará una interfaz de usuario clara y funcional, accesible desde cualquier navegador, que simplifique la consulta y visualización de los datos.

- **Implementación de Filtros Avanzados:** Se desarrollarán filtros personalizables, incluyendo rango de fechas y nivel de severidad (por ejemplo, para registros de vulnerabilidades), que permitan refinar las búsquedas y enfocar la información obtenida.
- **Visualización Estructurada de Resultados:** Los datos obtenidos y filtrados se presentarán de forma estructurada y comprensible para el usuario final, facilitando su interpretación rápida.
- **Exportación Flexible de Datos:** Se permitirá la descarga de los resultados en formatos reutilizables como JSON y CSV, optimizados para su posterior análisis o integración en otros sistemas.
- **Mejora y Adaptación de Herramienta Existente:** Se reutilizará y mejorará la base de una herramienta de consulta de datos del NIST desarrollada previamente en un entorno de consola, adaptándola a una aplicación web moderna y accesible para ampliar su utilidad y alcance.

Con el desarrollo de este proyecto, se pretende facilitar y agilizar la obtención y el manejo de datos técnicos estandarizados, ofreciendo una interfaz de usuario amigable que permita usar de manera sencilla todas las funciones de la aplicación.

## Descripción técnica

El proyecto **NistDataConverter** se desarrollará como una aplicación web de arquitectura cliente-servidor. El backend se encargará de la conexión con la API del NIST, del procesamiento de datos y de la generación de archivos para exportación. El frontend proporcionará una interfaz clara y funcional que permitirá al usuario realizar consultas, aplicar filtros y visualizar los resultados.

### Tecnologías a emplear:

- **Backend:**

#### Python (Flask o FastAPI):

Python es un lenguaje versátil y ampliamente utilizado en entornos académicos y profesionales. Flask y FastAPI son frameworks ligeros, ideales para construir APIs REST de forma rápida y clara. En este caso, **FastAPI** se valora por su rendimiento y su documentación automática integrada (Swagger).

#### Pandas:

Librería clave para manipulación de datos en Python. Facilita la conversión de estructuras complejas a **CSV**, lo cual es esencial en este proyecto.

- **Frontend:**

### **HTML, CSS y JavaScript:**

Forman el núcleo del desarrollo web. Se utilizarán para crear una interfaz de usuario funcional, sin necesidad de frameworks complejos. **JavaScript** permitirá dinamismo básico y conexión con el backend a través de `fetch`.

### **Bootstrap (opcional):**

Facilita el diseño responsive sin necesidad de crear estilos desde cero. Es útil para entregar una interfaz profesional en poco tiempo.

## **Arquitectura de la aplicación (por capas)**

La aplicación se estructurará en 3 capas principales:

### **1. Capa de presentación (frontend)**

- HTML, CSS, JavaScript
- Bootstrap (opcional)
- Formularios de búsqueda y botones de exportación

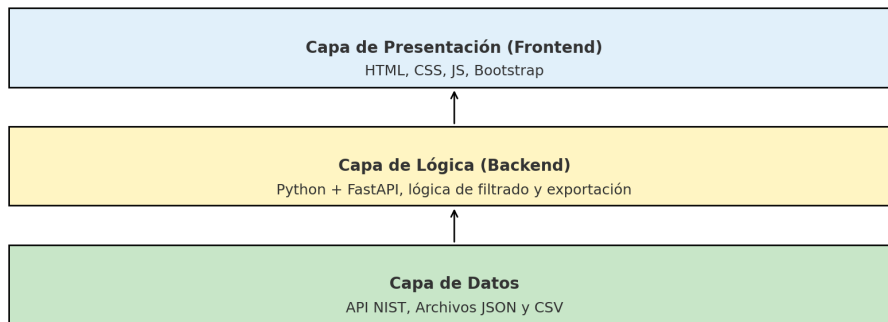
### **2. Capa de lógica (backend / servidor)**

- FastAPI en Python
- Módulo de conexión a la API del NIST
- Lógica de filtrado y exportación

### **3. Capa de datos**

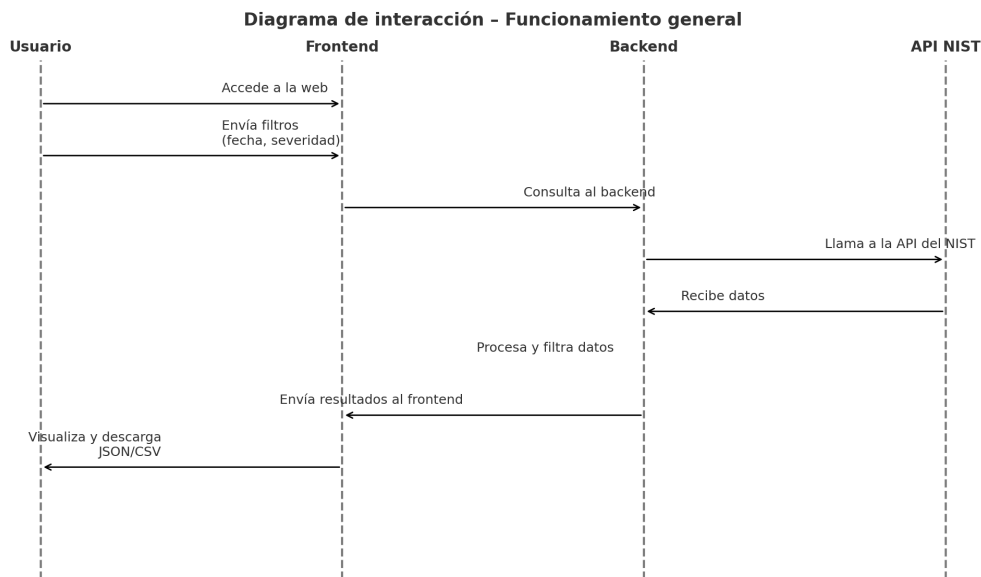
- API pública del NIST (fuente externa)
- Archivos generados (JSON / CSV)
- No requiere base de datos local

### Arquitectura de NistDataConverter



### Funcionamiento general:

1. El usuario accede a la interfaz web.
2. Introduce parámetros de búsqueda o aplica filtros (fecha, severidad).
3. El frontend envía la consulta al backend.
4. El backend se comunica con la API del NIST y obtiene los datos.
5. Los datos se procesan, filtran y devuelven al frontend para su visualización.
6. El usuario puede optar por descargar los resultados en formato JSON o CSV.



## Proceso de desarrollo a utilizar en el proyecto

Para el desarrollo del proyecto **NistDataConverter** se utilizará una **metodología incremental basada en entregas iterativas**. Esta metodología permite desarrollar el sistema por fases, construyendo versiones funcionales parciales que van incorporando nuevas funcionalidades de forma progresiva. Esta opción es adecuada para un entorno académico, donde se dispone de un tiempo limitado y se requiere mantener un control continuo del progreso.

El enfoque incremental es más flexible que el modelo en cascada tradicional, ya que permite validar cada parte del sistema antes de avanzar, detectar errores a tiempo y corregir desviaciones sin necesidad de reestructurar el proyecto completo. También se alinea con la práctica habitual en proyectos modernos de desarrollo ágil..

### Fases del desarrollo:

#### 1. Análisis de requisitos y planificación

- Estudio detallado de la API pública del NIST.
- Definición de funcionalidades prioritarias y filtros disponibles (fecha y severidad).
- Planificación temporal del desarrollo y preparación del entorno de trabajo.

#### 2. Diseño del sistema y de la interfaz

- Diseño de la arquitectura cliente-servidor (frontend, backend, API).
- Selección de tecnologías y herramientas concretas para cada capa.

### **3. Primera iteración – Backend básico y conexión API**

- Implementación de la conexión con la API del NIST.
- Procesamiento y recepción de los datos en formato JSON.
- Desarrollo de una versión inicial del backend con FastAPI.

### **4. Segunda iteración – Interfaz web y filtros funcionales**

- Desarrollo del frontend básico con HTML, CSS y JS.
- Integración de los formularios de búsqueda y filtros por fecha/severidad.
- Visualización de resultados obtenidos del backend.

### **5. Tercera iteración – Exportación de datos y mejoras UI**

- Implementación de la descarga de datos en formato JSON y CSV usando Pandas.
- Mejora visual del frontend con Bootstrap.
- Validación de entradas y manejo de errores en la interfaz.

### **6. Pruebas del sistema y validación final**

- Pruebas funcionales de extremo a extremo.
- Verificación de filtros, consultas y exportaciones.
- Revisión del comportamiento con distintos navegadores.

### **7. Despliegue online del proyecto**

- Configuración y subida del proyecto a una plataforma gratuita como Render o Replit.
- Pruebas del sistema desplegado en entorno real.
- Compartición del enlace para la demostración final.

## 8. Documentación del proyecto y presentación

- Redacción de la memoria técnica completa del proyecto.
- Creación del manual de usuario con capturas.
- Preparación de la presentación para la defensa final.

### Planificación do traballo e estimación temporal

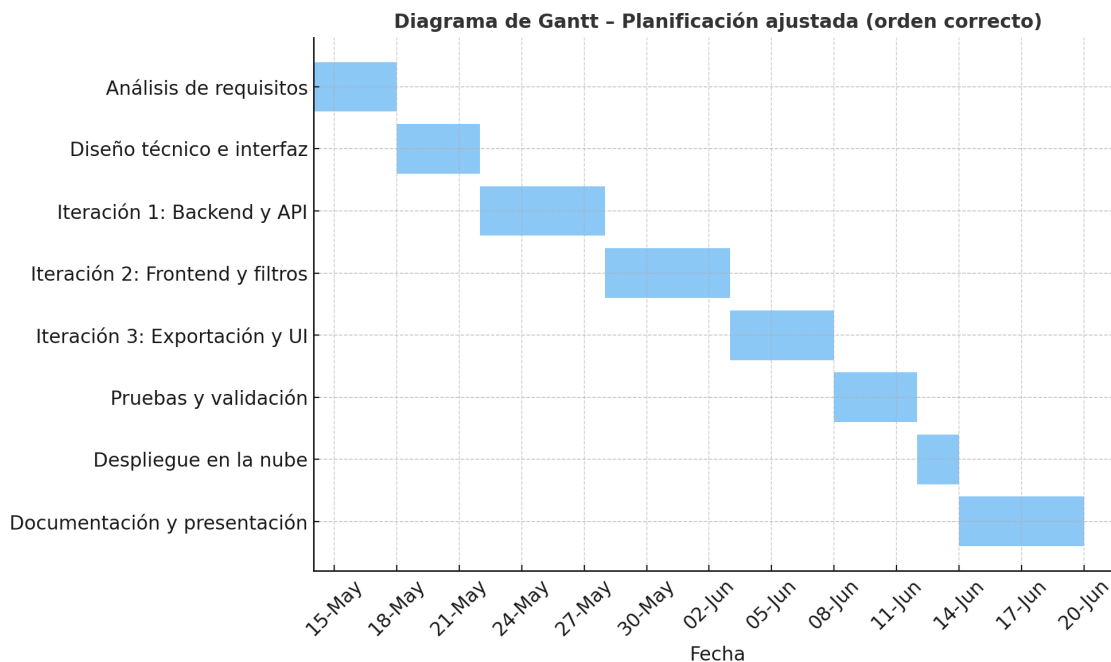
El proyecto NistDataConverter se desarrollará siguiendo una metodología incremental basada en entregas iterativas, con una dedicación prevista y una estimación temporal detallada para cada fase.

Dedicación semanal prevista: **30 horas/semana** Dedicación diaria prevista: **6 horas/día** (Considerando 5 días laborables por semana, 30 horas / 5 días = 6 horas/día)

Dedicación semanal prevista (en horas/semana):	
Fase	Estimación temporal (en días)
Análisis de requisitos	4 días
Diseño técnico e interfaz	4 días
Iteración 1: Backend y conexión con API	6 días
Iteración 2: Frontend y filtros	6 días
Iteración 3: Exportación y mejoras UI	5 días
Pruebas y validación	4 días
Despliegue en la nube	2 días
Documentación y presentación	6 días
<b>TOTAL PROXECTO</b>	<b>37 días</b>

Nombre tarea	Estimación (horas)	Fecha inicio	Fecha fin
Análisis de requisitos	24	2025-05-30	2025-06-04
Diseño técnico e interfaz	24	2025-06-05	2025-06-10
Iteración 1: Backend y conexión API	36	2025-06-11	2025-06-18
Iteración 2: Frontend y filtros	36	2025-06-19	2025-06-26
Iteración 3: Exportación y mejoras UI	30	2025-06-27	2025-07-03
Pruebas y validación	24	2025-07-04	2025-07-09
Despliegue en la nube	12	2025-07-10	2025-07-11

Documentación y presentación	36	2025-07-14	2025-07-21
<b>TOTAL PROYECTO</b>	<b>222</b>		



## Descripción da documentación a entregar no proxecto

### Memoria Técnica

- **Introducción:** Propósito del proyecto NistDataConverter y objetivos clave.
- **Análisis y Diseño:** Requisitos funcionales (historias de usuario), arquitectura cliente-servidor, diagramas de flujo y tecnologías empleadas (Python/FastAPI, HTML/CSS/JS).
- **Proceso de Desarrollo:** Metodología incremental, fases y planificación temporal (incluyendo tabla y diagrama de Gantt).
- **Pruebas y Despliegue:** Estrategias de prueba, verificación de funcionalidades y proceso de puesta en producción.
- **Conclusiones y Futuras Mejoras:** Balance del proyecto y posibles evoluciones.
- **Medios y Bibliografía:** Hardware, software utilizados y referencias consultadas.

### Manual de Usuario

- **Guía Rápida:** Instrucciones para acceder y utilizar la aplicación NistDataConverter.
- **Funcionalidades:** Cómo realizar consultas, aplicar filtros y exportar datos en JSON/CSV.



- **Ayuda:** Sección de preguntas frecuentes y resolución de problemas.
- **Contenido visual:** Inclusión de capturas de pantalla para facilitar el aprendizaje.

### Código Fuente

- **Repositorio Git/GitHub:** Acceso al código backend (Python/FastAPI) y frontend (HTML/CSS/JS) con su estructura y organización.

## Medios materiales necesarios

- **Hardware:**
  - Portátil para el desarrollo de la aplicación web, pruebas locales y despliegue online:
    - **Modelo:** ASUS ZenBook Duo 14
    - **Procesador:** Intel Core i7-1165G7
    - **RAM:** 16GB
    - **Disco duro:** SSD NVMe de 512GB
    - **Sistema operativo:** Arch Linux (EndeavourOS)
    - **Conexión a Internet** estable para el consumo de la API y el despliegue en la nube
    - **Coste estimado:** 1.300 € (equipo ya en propiedad)

### Uso en el proyecto:

Este equipo se utilizará en **todas las fases del desarrollo**: análisis, programación, pruebas, documentación y despliegue.

- **Software:**

Software / Plataforma	Versión / Tipo	Coste	Uso en el proyecto
<b>Sistema operativo</b>	Arch Linux (EndeavourOS)	Gratuito	Todas las fases
<b>IDE</b>	Visual Studio Code v1.89	Gratuito	Desarrollo backend y frontend
<b>Control de</b>	Git 2.44 + GitHub	Gratuito	Control de código y versiones en

<b>versiones</b>			todas las fases
<b>Navegador web</b>	Brave v1.65	Gratuito	Pruebas frontend y navegación API (pruebas, validación)
<b>Plataformas de despliegue</b>	Render / Replit / Railway	Gratuitas (planes básicos)	Fase de despliegue y demostración final

## Bibliografía

Flask. (s.f.). *Flask Documentation (2.x)*. Flask Project. Recuperado de: <https://flask.palletsprojects.com/>

FastAPI. (s.f.). *FastAPI - Modern web framework for Python*. Tiangolo. Recuperado de: <https://fastapi.tiangolo.com/>

Pandas. (s.f.). *The pandas library*. The Pandas Project. Recuperado de: <https://pandas.pydata.org/>

Python Software Foundation. (s.f.). *Python 3 Documentation*. Recuperado de: <https://docs.python.org/3/>

Bootstrap. (s.f.). *Bootstrap 5 Documentation*. The Bootstrap Team. Recuperado de: <https://getbootstrap.com/>

NIST. (s.f.). *NIST API Developer Resources*. U.S. National Institute of Standards and Technology. Recuperado de: <https://www.nist.gov/developer>

Render. (s.f.). *Render Docs*. Recuperado de: <https://render.com/>

Replit. (s.f.). *Replit Documentation*. Recuperado de: <https://replit.com/>

Railway. (s.f.). *Railway Docs*. Recuperado de: <https://railway.app/>

## Observaciones

Durante el desarrollo de este anteproyecto se ha hecho uso de herramientas de inteligencia artificial (IA) de forma complementaria para:

- Apoyar la redacción y estructuración del documento.
- Generar diagramas de arquitectura e interacción.
- Obtener ejemplos de código y validar su funcionamiento.
- Revisar y ajustar el formato de apartados como la bibliografía.

Estas herramientas se han empleado con fines educativos y de apoyo, manteniendo siempre el criterio propio del autor en el diseño, desarrollo y justificación técnica del proyecto.

La utilización de IA se ajusta a las directrices actuales del centro y se menciona de forma explícita en este documento como parte del compromiso con la transparencia y la ética académica.