**UNIDAD NACIONAL DE PROTECCIÓN - UNP**

**INFORME GRUPO DE GESTIÓN DE LAS TECNOLOGÍAS DE LA INFORMACIÓN**

**-GGTI-**

**ECOSISTEMA UNP – VERSION 1.0**

****

HERRAMIENTAS DE ANALISIS Y TESTEO

CIBERSEGURIDAD

Test. V.1

**Pruebas Estáticas – Código Certificados Laborales**

Bogotá, 5 de noviembre del 2024

**CONTENIDO**

|  |  |  |
| --- | --- | --- |
| [Nomenclatura] | [Tema] | [Página] |
| I | Identificación parcial de Riesgos relacionados con los sistemas de información “Sistema de gestión de seguridad de la información (SGSI) – Resumen Ejecutivo | 5 |
| II | Aplicación de pruebas (Análisis de Riesgos de Ciberseguridad – Ciberataques ) | 5 |
| III | Subsanación de Brechas o sugerencias de corrección | 12 |

1. **PROPÓSITO**

* Identificación parcial de Riesgos relacionados con los sistemas de información “Sistema de gestión de seguridad de la información (SGSI)”. Ecosistema UNP
* Aplicación de pruebas estáticas (Análisis de Riesgos de Ciberseguridad – Ciberataques )
* Subsanación de Brechas o sugerencias de corrección.

1. **ALCANCE**
2. Certificaciones Laborales
3. Dependencias o Misiones UNP
4. Personal Involucrado Directo e Indirecto
5. **DEFINICIONES**

### **Análisis Estático**

Estas herramientas examinan el código fuente en busca de patrones que puedan indicar vulnerabilidades, errores lógicos y malas prácticas de codificación. No se requiere la ejecución del programa para realizar este análisis.

### **Detección de Vulnerabilidades Comunes**

Identifican problemas de seguridad como inyecciones de SQL, desbordamientos de buffer, uso inseguro de funciones, entre otras. Esto es crucial para asegurar que el sistema de gestión de la información (SGI) cumpla con los requisitos de la norma ISO/IEC 27001:2022, que incluye la identificación y tratamiento de riesgos de seguridad.

### **Mejora de la Calidad del Código**

Además de encontrar fallos de seguridad, también ayudan a mantener la calidad del código asegurando que siga las mejores prácticas y estándares de codificación. Esto es parte del proceso de gestión de riesgos y mejora continua que promueve la norma ISO/IEC 27001:20221.

1. **ASPECTOS RELEVANTES**

### Se toma como referencia la normas ISO 27031 del 2011, ISO 27032 / 2022, ISO 27001 / 2022 y RESOLUCION 1774 / 2024 en Relación con la normativas y lineamientos de Gobierno MINTIC.

La Triada de Seguridad y del Sistema: Confidencialidad – Integridad - Disponibilidad

 **Continuidad del Negocio**: Ambas fuentes destacan la relevancia de mantener la operatividad en situaciones de emergencia. ISO 27031 ofrece un marco para la planificación y gestión de la continuidad, mientras que la Resolución 1774 establece pautas concretas para asegurar esta continuidad en el ámbito cibernético. La norma ISO/IEC 27001 también es fundamental, ya que establece los requisitos para un sistema de gestión de la seguridad de la información que contribuye a la continuidad del negocio.

 **Evaluación de Riesgos**: Tanto la norma como la resolución subrayan la importancia de realizar evaluaciones de riesgos para detectar vulnerabilidades y amenazas, lo que permite a las organizaciones estar mejor preparadas y responder de manera efectiva. ISO 27001 refuerza esta práctica al exigir un enfoque sistemático para la identificación y evaluación de riesgos.

 **Capacitación y Concienciación**: Ambas iniciativas destacan la necesidad de capacitar al personal en temas de ciberseguridad y continuidad del negocio, garantizando que todos los niveles de la organización estén informados y listos para actuar. La ISO 27001 también incluye requisitos específicos sobre la formación y concienciación del personal en relación con la seguridad de la información.

 **Pruebas y Ejercicios**: La Resolución 1774 menciona la importancia de llevar a cabo pruebas regulares de los planes de continuidad, un aspecto crucial que se aborda en ISO 27031. La norma ISO 27001, por su parte, también promueve la realización de pruebas para asegurar que los controles de seguridad son efectivos y se ajustan a los requisitos establecidos.

 **Mejora Continua**: Tanto ISO 27031 como la Resolución 1774 fomentan la mejora constante en las prácticas de gestión de riesgos y continuidad, adaptándose a nuevas amenazas y cambios en el entorno operativo. ISO 27001 refuerza este enfoque al exigir que las organizaciones implementen un ciclo de mejora continua en su sistema de gestión de la seguridad de la información.

 **Cumplimiento de la Resolución 1774 de 2024**: Emitida por la Agencia Nacional de Seguridad Informática (ANSI), esta resolución establece requisitos y directrices específicas para la gestión de la seguridad de la información en organizaciones, especialmente en los sectores público y privado. Al combinarla con la norma ISO/IEC 27001:2022, se puede crear un marco sólido para la gestión de la seguridad de la información y otras normativas relacionadas, garantizando una integración efectiva de las mejores prácticas en ciberseguridad.

1. **CONTENIDO.**

#### **Resumen Ejecutivo**

**Herramienta 1.** Bandit

* **Resumen de Hallazgos**: Usar herramientas de análisis estático que busca vulnerabilidades de seguridad en el código Python.
* **Información del análisis:**
* **Tests incluidos/excluidos**: No se incluyeron ni excluyeron tests específicos.
* **Versión de Python**: Python 3.13.0
* **Fecha y hora de inicio**: 2024-11-04 22:09:23

**Resultados del Test:**

1. **[B311:blacklist] Generadores pseudoaleatorios estándar no son adecuados para propósitos de seguridad/criptografía**:
   * **Severidad**: Baja
   * **Confianza**: Alta
   * **CWE**: CWE-330 (https://cwe.mitre.org/data/definitions/330.html)
   * **Más información**: [Bandit B311](https://bandit.readthedocs.io/en/1.7.10/blacklists/blacklist_calls.html?form=MG0AV3#b311-random)
   * **Ubicación**: Varias líneas en certificacione.py
   * **Descripción**: Uso de random.randint y random.choice que no son seguros para fines criptográficos.
2. **[B105:hardcoded\_password\_string] Posible contraseña codificada**:
   * **Severidad**: Baja
   * **Confianza**: Media
   * **CWE**: CWE-259 (https://cwe.mitre.org/data/definitions/259.html)
   * **Más información**: [Bandit B105](https://bandit.readthedocs.io/en/1.7.10/plugins/b105_hardcoded_password_string.html?form=MG0AV3)
   * **Ubicación**: settings.py, línea 24
   * **Descripción**: Hay una clave secreta codificada SECRET\_KEY que debería mantenerse segura.

**Métricas del Análisis:**

* **Total de líneas de código**: 1261
* **Total de líneas omitidas**: 0
* **Problemas encontrados**:
  + **Por severidad**: 4 (baja), 0 (media), 0 (alta)
  + **Por confianza**: 3 (alta), 1 (media), 0 (baja)
* **Archivos omitidos**: 0
* **Conclusiones y Recomendaciones**:
* Resumen de las principales conclusiones y recomendaciones de mitigación.
* **Reemplazar generadores pseudoaleatorios**: Usa el módulo secrets en lugar de random para tareas de seguridad:

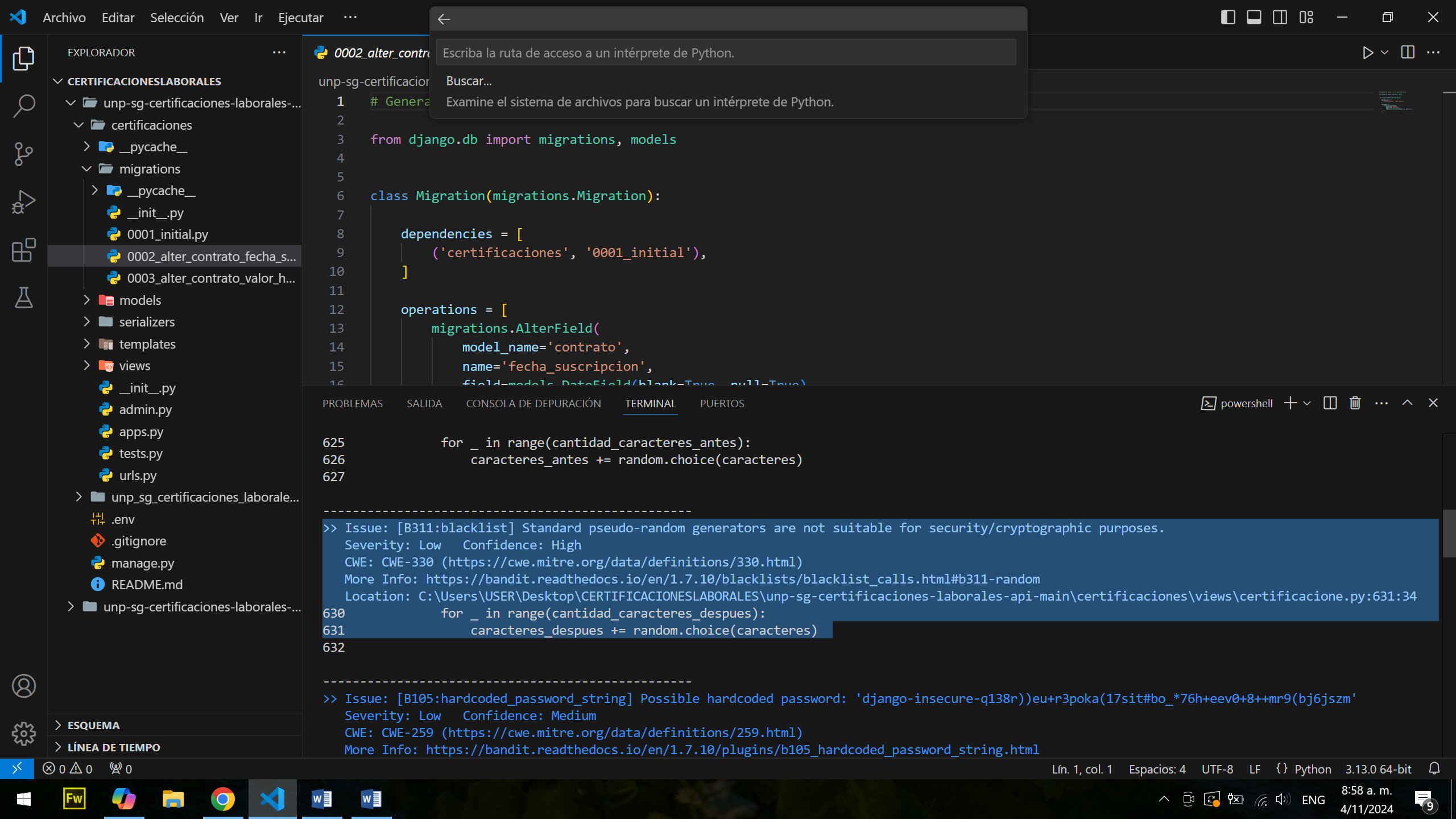
python

import secrets

cantidad\_caracteres\_antes = secrets.randbelow(10 - longitud\_numero) + 1

caracteres\_antes += secrets.choice(caracteres)

* **Mover claves secretas a variables de entorno**: Las contraseñas y claves secretas no deben estar codificadas en el código fuente. Mueve SECRET\_KEY a una variable de entorno.



**Herramienta 2.** PyUp

* **Resumen de Hallazgos**: Informe detallado sobre las dependencias desactualizadas y las vulnerabilidades conocidas en el código.
* **Información del análisis:**

Este reporte indica varios problemas:

1. **Advertencia de configuración de base de datos**:
2. WARNING | ln:132 - No database configured...
3. **Error de subida**:

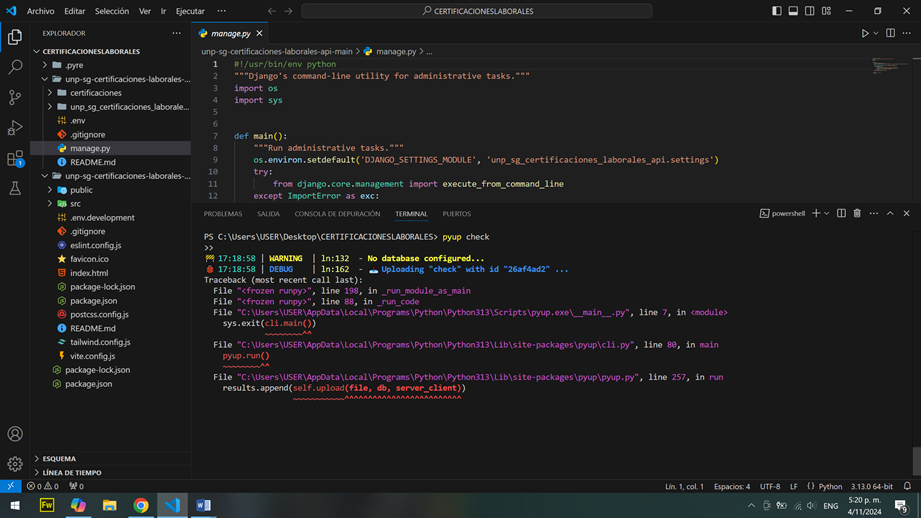
DEBUG | ln:162 - 📤 Uploading "check" with id "a01bc13b" ...

**Métricas del Análisis:**

* **Total de líneas de código**: 1261
* **Total de líneas omitidas**: 0

**Conclusiones y Recomendaciones**:

* Realizar pruebas Dinámicas de Ciberseguridad una vez conectada la BD.



**Herramienta 3. NPM Audit**

**Resumen de Hallazgos**: herramienta de línea de comandos incluida con NPM que te ayuda a identificar y corregir vulnerabilidades en las dependencias de tus proyectos.

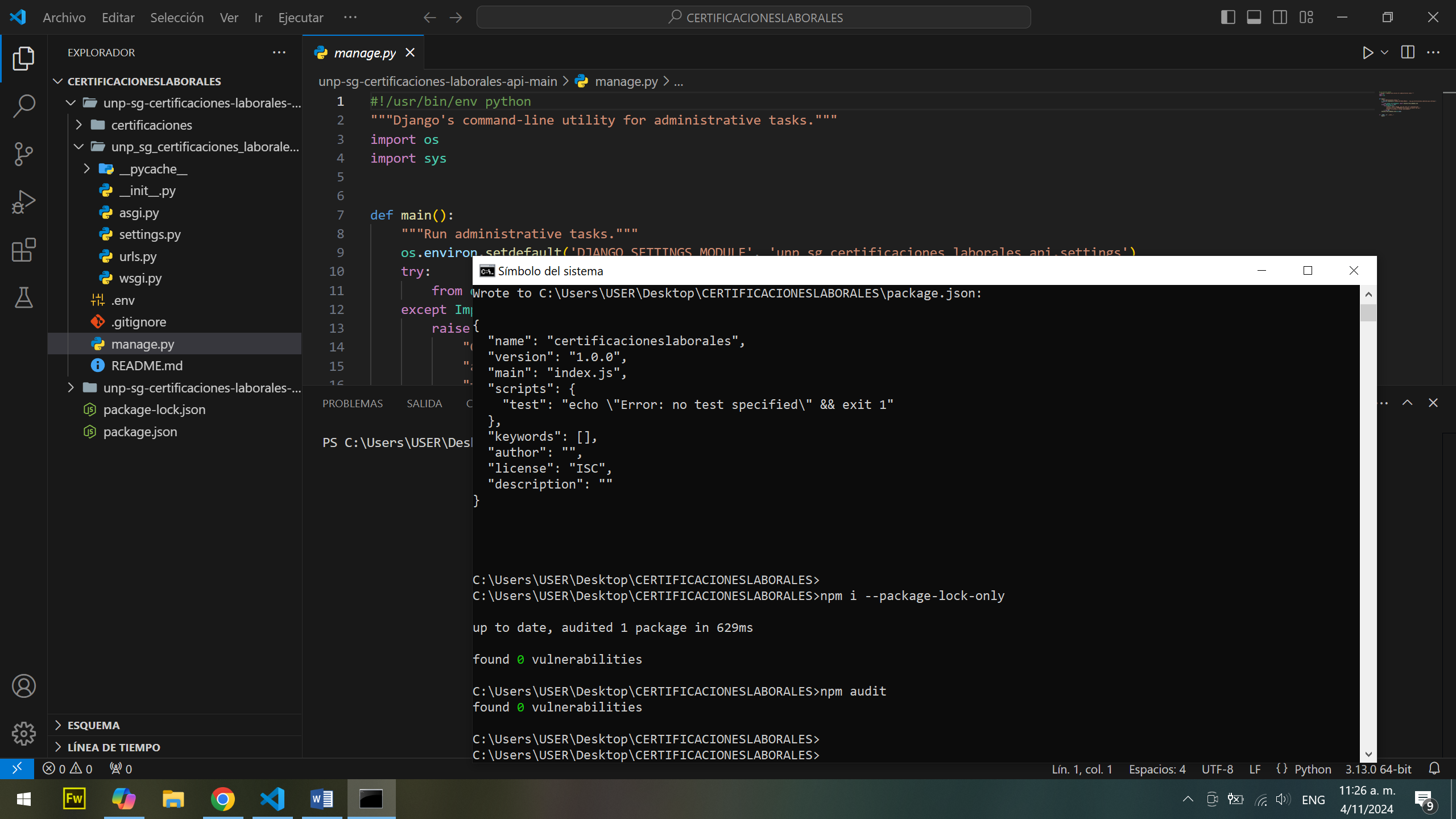
* **Información del análisis:**
* "código resiliente"

**Métricas del Análisis:**

* **Total de líneas de código**: 1261
* **Total de líneas omitidas**: 0

**Conclusiones y Recomendaciones**:

* NPM no identifica Vulnerabilidades



* **Herramienta 4.** Flake8
* **Resumen de Hallazgos**: **Código limpio y legible**: Mantener un código bien estructurado y fácil de leer es crucial para detectar y mitigar vulnerabilidades. El código limpio facilita las revisiones y auditorías de seguridad
* **Ítem. 8.2 anexo A Iso 27002**

**Información del análisis:**

* "código con redundancia"

### Detalle del Análisis - código estructurado

1. **F811: redefinición de 'admin' no utilizado**
   * **Archivo y línea**: admin.py:12
   * **Descripción**: admin se redefine sin ser utilizado desde la línea 1. Asegúrate de que no estás importando el mismo módulo o variable más de una vez.
2. **F401: 'django.urls.path' importado pero no utilizado**
   * **Archivo y línea**: admin.py:17
   * **Descripción**: path está importado pero no se utiliza en el código. Elimina las importaciones innecesarias para limpiar tu código.
3. **F401: 'django.utils.html.format\_html' importado pero no utilizado**
   * **Archivo y línea**: admin.py:18
   * **Descripción**: format\_html está importado pero no se utiliza en el código. Elimina las importaciones innecesarias.
4. **E302: se esperaban 2 líneas en blanco, se encontró 1**
   * **Archivo y líneas**: admin.py:20, admin.py:24
   * **Descripción**: Según las convenciones de estilo de Python (PEP 8), se requieren 2 líneas en blanco antes de las definiciones de funciones o clases.
5. **E501: línea demasiado larga**
   * **Varias líneas en** admin.py **(por ejemplo, 25, 32, 33, 41, 42, etc.)**
   * **Descripción**: Las líneas de código tienen más de 79 caracteres. Divide las líneas largas para mejorar la legibilidad del código.
6. **W293: línea en blanco contiene espacios en blanco**
   * **Archivo y línea**: admin.py:61
   * **Descripción**: Elimina los espacios en blanco de las líneas en blanco para cumplir con PEP 8.
7. **F841: variable local 'contrato' asignada pero no utilizada**
   * **Archivo y línea**: admin.py:96
   * **Descripción**: La variable contrato se asigna pero no se usa. Considera eliminarla o utilizarla en el código

**Recomendaciones**

* **Eliminar Importaciones Innecesarias**: Revisa tus importaciones y elimina las que no estés utilizando.

python

from django.urls import path # Si no se usa, elimina esta línea.

* **Ajustar Longitud de las Líneas**: Divide las líneas largas para que no excedan los 79 caracteres.

python

# Línea larga:

if ejemplo\_muy\_largo and otra\_condicion\_muy\_larga and una\_tercera\_condicion\_aun\_mas\_larga:

# Línea ajustada:

if (ejemplo\_muy\_largo and otra\_condicion\_muy\_larga and

una\_tercera\_condicion\_aun\_mas\_larga):

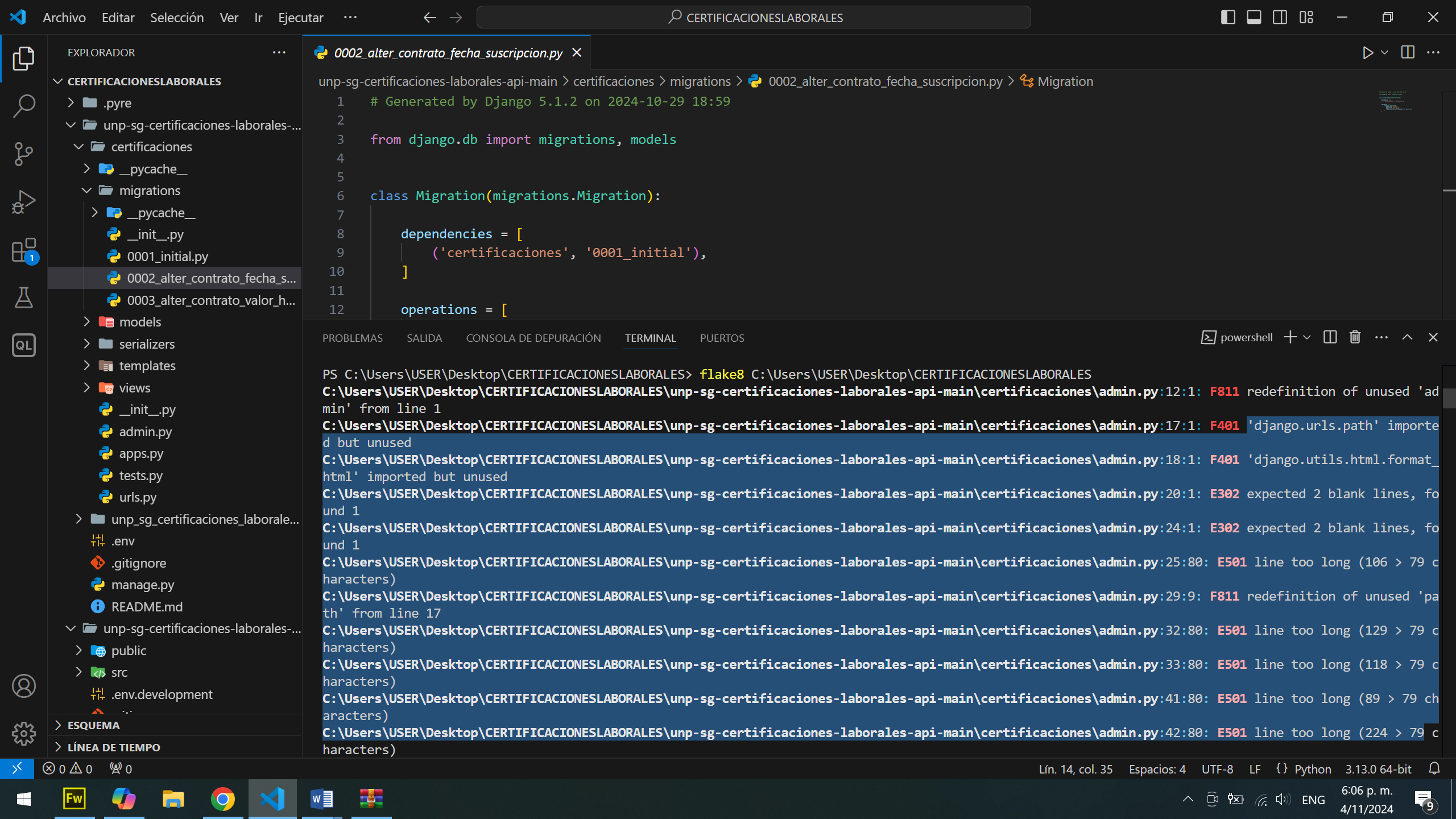
* **Eliminar Variables No Utilizadas**: Revisa las variables y elimina las que no se utilizan.

python

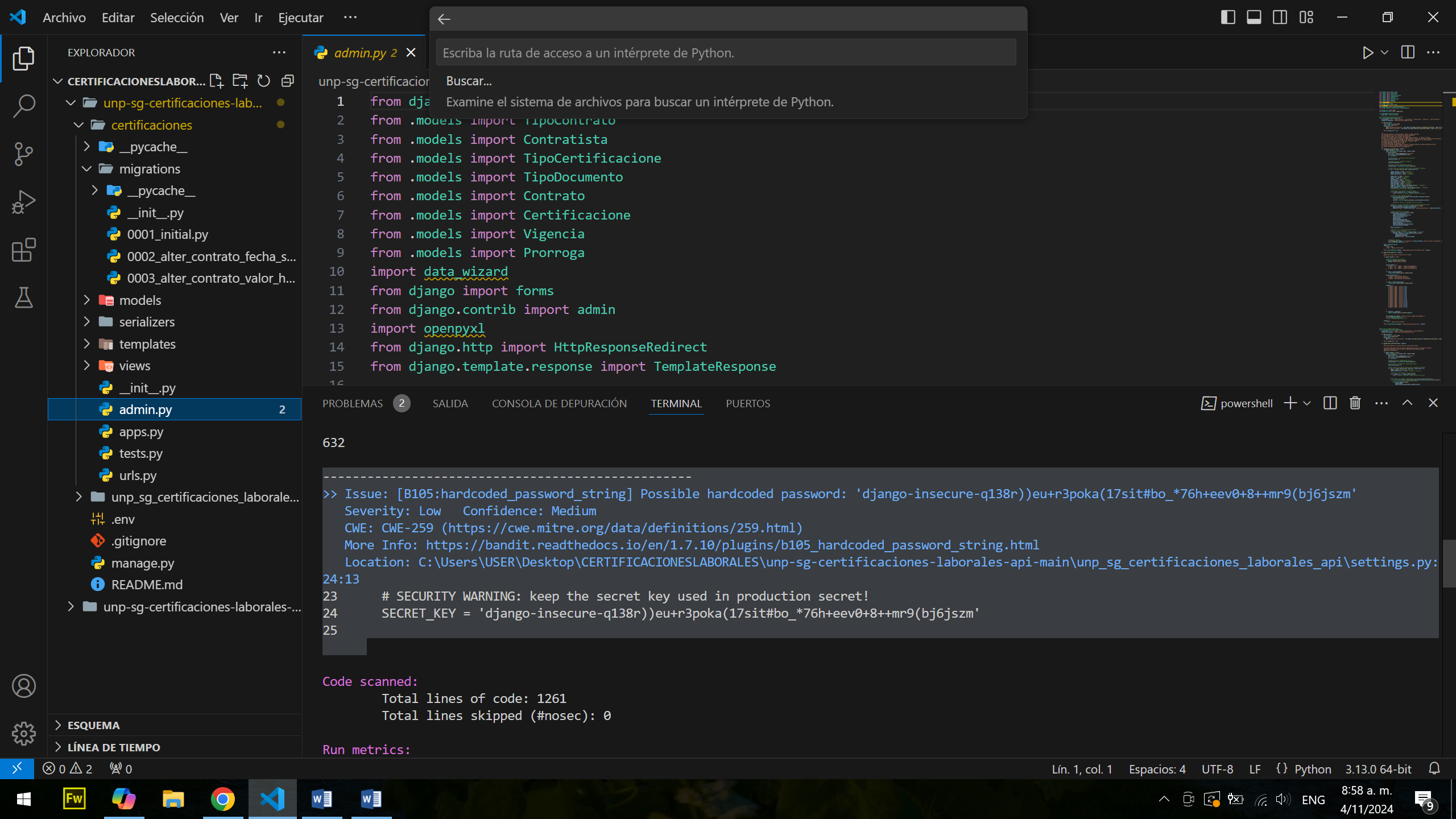
contrato = "valor" # Si no se usa, elimina esta línea.

**Conclusiones y Recomendaciones**:

"Debido a la estructura robusta del código y a la complejidad derivada de sus dependencias, librerías, API o framework utilizado, no es posible identificar todas sus limitaciones de manera inmediata. Por lo tanto, se recomienda realizar pruebas dinámicas enfocadas en la ciberseguridad para asegurar que no existan problemas no detectados."



#### **Metodología**

* **Herramientas Utilizadas**: Herramientas de análisis estático utilizadas (Bandit, **NPM Audit,** PyUp, SonarQube, Flake8).
* **Proceso de Análisis**:
* 

#### **Hallazgos Detallados**

* **Vulnerabilidad 1**:

**Descripción**: **[B311:blacklist] Generadores pseudoaleatorios estándar no son adecuados para propósitos de seguridad/criptografía:** random.randint y random.choice

* + **Severidad**: Baja
  + **Confianza**: Alta
  + **Recomendación**: Descritas anteriormente.

**Vulnerabilidad 2**:

**Descripción**: **[B105:hardcoded\_password\_string] Posible contraseña codificada [B105:hardcoded\_password\_string] Posible contraseña codificada**: clave secreta codificada SECRET\_KEY

* + **Severidad**: Baja
  + **Confianza**: Media
  + **Ubicación**: settings.py, línea 24

#### **Análisis por Categoría**

* **Inyecciones SQL**: Vulnerabilidades relacionadas con inyecciones SQL. (No Aplica)
* **Desbordamientos de Buffer**: Vulnerabilidades relacionadas con desbordamientos de buffer. (No Aplica)
* **Uso de Funciones Inseguras**: Vulnerabilidades relacionadas con el uso inseguro de funciones. (Aplica)

#### **Métricas y Estadísticas**

* **Total de Vulnerabilidades Encontradas**: Directas 2 (sujetas a cambio o afirmación)
* **Distribución por Severidad**: Gráfico o tabla que muestre la distribución de las vulnerabilidades por nivel de severidad (baja, media, alta). ( N/A)
* **Distribución por Categoría**: Gráfico o tabla que muestre la distribución de las vulnerabilidades por categoría. (N/A)

#### **Recomendaciones Generales**

Bajo **herramientas estáticas de verificación de código** una aplicación es utilizada para analizar y revisar el código fuente de un programa sin necesidad de ejecutarlo. Estas herramientas son esenciales en el ámbito de la ciberseguridad, ya que ayudan a identificar vulnerabilidades y problemas de seguridad antes de que el software se despliegue.

### Importancia en Ciberseguridad

Las herramientas estáticas de verificación de código permiten detectar problemas de seguridad de manera anticipada, lo cual es crucial para mitigar riesgos y asegurar que el código sea robusto y seguro. Al identificar vulnerabilidades como inyecciones SQL, desbordamientos de buffer y uso inseguro de funciones, estas herramientas aseguran que el código cumpla con los estándares de seguridad y buenas prácticas, contribuyendo a la integridad y confiabilidad del software.

### Beneficios en el Desarrollo In-House

1. **Detección Temprana de Errores**: Identificar problemas antes del despliegue reduce los costos asociados con la corrección de errores en etapas posteriores del ciclo de vida del software.
2. **Mejora de la Calidad del Código**: Al seguir convenciones de estilo y detectar problemas comunes, se mejora la legibilidad y mantenibilidad del código.
3. **Cumplimiento de Normativas**: Ayuda a asegurar que el código cumpla con normativas de seguridad, como la ISO/IEC 27001:2022, que exige la identificación y tratamiento de riesgos de seguridad.

### Buenas Prácticas de Programación

Para maximizar los beneficios de las herramientas estáticas de verificación de código, es importante seguir buenas prácticas de programación, especialmente en el desarrollo in-house:

1. **Revisión de Código**: Realizar revisiones de código regulares para identificar y corregir problemas antes de que se conviertan en vulnerabilidades.
2. **Documentación**: Mantener una documentación clara y detallada del código, lo que facilita la comprensión y mantenimiento por parte de otros desarrolladores.
3. **Pruebas Unitarias**: Implementar pruebas unitarias para verificar que cada componente del código funcione correctamente de manera aislada.
4. **Automatización**: Integrar herramientas estáticas de verificación de código en el flujo de trabajo de integración continua (CI/CD) para análisis automáticos con cada commit o despliegue.
5. **Capacitación Continua**: Fomentar la capacitación continua del equipo de desarrollo en las últimas tendencias y prácticas de ciberseguridad.
6. **Gestión de Dependencias**: Utilizar herramientas como Dependabot o Snyk para mantener las dependencias de software actualizadas y libres de vulnerabilidades.

#### **Conclusiones**

* **Resumen de los Resultados**: verificar si las fallas detectadas son congruentes con el código y/o subsanar.
* **Pasos a Seguir**: Ajustar cambios en el código de ser necesarios y/o documentar porque son o no relativos para mitigar las vulnerabilidades y mejorar la seguridad del código.
* **La norma ISO/IEC 27001:2022** establece un sistema de gestión de seguridad de la información (SGSI) y exige la identificación y tratamiento de riesgos de seguridad. Integrar buenas prácticas de programación con las herramientas estáticas de verificación de código asegura que el software no solo sea seguro, sino que también cumpla con los estándares internacionales de seguridad.

#### **Anexos**

* **Detalles Técnicos**: Información técnica adicional, como scripts utilizados durante el análisis. (No Aplica)
* **Logs de Herramientas**: Registros detallados de las herramientas utilizadas durante el análisis. (Incorporados en imágenes)
* **Referencias**: Cualquier referencia o recurso adicional consultado durante el análisis. (No Aplicado)

1. Situación analizada y gestión realizada:

Definido por los tiempos y los cambios de ejecución (sujetos a Despliegue / Procesos / Pruebas/ implementación de Normas y Estándares de Calidad, etc.).

1. Riesgos asociados: [Opcional]

* []

1. Alternativas de solución planteadas: [Opcional]

* []

1. Plan De Acción: [Opcional]

| **No.** | **Acción de mejora** | **Responsable y plazo** |
| --- | --- | --- |
|  |  |  |
|  |  |  |

1. **ANEXOS.**
2. [] [NO APLICA]

**Cordialmente,**

**[JEAVER ALEXANDER VERGARA FERNANDEZ].**

[CIBERSEGURIDAD - ECOSISTEMA]

[Grupo GGT]

[OAPI]

Unidad Nacional de Protección

|  |  |  |
| --- | --- | --- |
| **GET-FT-05 /V1** | **Oficialización: 15/07/2015** | Página 1 de 5 |

**INSTRUCTIVO**

Pautas para la elaboración del Cuerpo del Formato:

Contienen los diferentes campos de información que deben ser diligenciados. Incluir la firma o firmas del responsable de la información consignada en él y la fecha del diligenciamiento del Formato Informe de Estado y Gestión..

El cuerpo del Informe debe incluir:

|  |  |  |
| --- | --- | --- |
| *CONTENIDO* | *FR* | *ASPECTOS A TENER EN CUENTA* |
| ENCABEZADO | X | Según las Pautas del numeral **5.4** de la Guía **SGI-GU-01/V4** |
| PROPÓSITO | X | Según la estructura del numeral **5.1.2** de la Guía **SGI-GU-01/V4** |
| ALCANCE | X | Según la estructura del numeral **5.1.2** de la Guía **SGI-GU-01/V4** |
| DEFINICIONES | X | Según la estructura del numeral **5.1.2** de la Guía **SGI-GU-01/V4** |
| ASPECTOS RELEVANTES | X | Relacionar los aspectos significativos que impactan el Estado y la Gestión Tecnológica |
| CONTENIDO: | X | El contenido del Informe debe describir cada uno de los temas y/o actividades desarrolladas, de manera concreta, incluyendo especificaciones técnicas de ser necesario.  Para cada uno de los eventos y/o actividades incluidas en el Informe, es necesario que se detalle:   1. Situación analizada y gestión realizada [X] 2. Riesgos asociados [Opcional] 3. Alternativas de solución planteadas [Opcional] 4. Plan de Acción [Opcional]   Diligenciar Formato:   | **No.** | **Acción de mejora** | **Responsable y plazo** | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  |   La información incluida por cada aspecto debe evidenciar las acciones específicas en materia de la Gestión Tecnológica que el personal del GGTI realiza como parte de las funciones asignadas, bajo los lineamientos o instrucciones impartidas. Para casos particulares es posible que requiera la gestión y participación de otras áreas. |
| Anexos | O |  |
| Firma | X |  |
| Créditos | X |  |

Requerido = X Opcional= O