

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA  
FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**PROYECTO FINAL DEL CURSO ASEGURAMIENTO DE CALIDAD DE  
SOFTWARE  
PRUEBAS DE SEGURIDAD**

Presentado por:

**OMAR ALEJANDRO BERMUDEZ CACERES**

**ALVARO VINICIO ORELLENA MENDEZ**

**CARLOS EDUARDO CAMEY MILIAN**

**HERBERT ARAGON MONZON**

**MAYCOL ARNULFO GUERRA**

**KEVIN MAURICIO PUMAY GODOY**

CONOCEREIS LA VERDAD  
Y LA VERDAD OS HARÁ LIBRES

VILLA NUEVA, OCTUBRE DE 2024

## Índice

<b>Introducción.....</b>	<b>3</b>
<b>Error Lens.....</b>	<b>4</b>
<b>Dotenv Official + Vault.....</b>	<b>7</b>
<b>Snyk Security – Code .....</b>	<b>8</b>
<b>Red Hat Dependency Analytics.....</b>	<b>13</b>
<b>Conclusión.....</b>	<b>19</b>

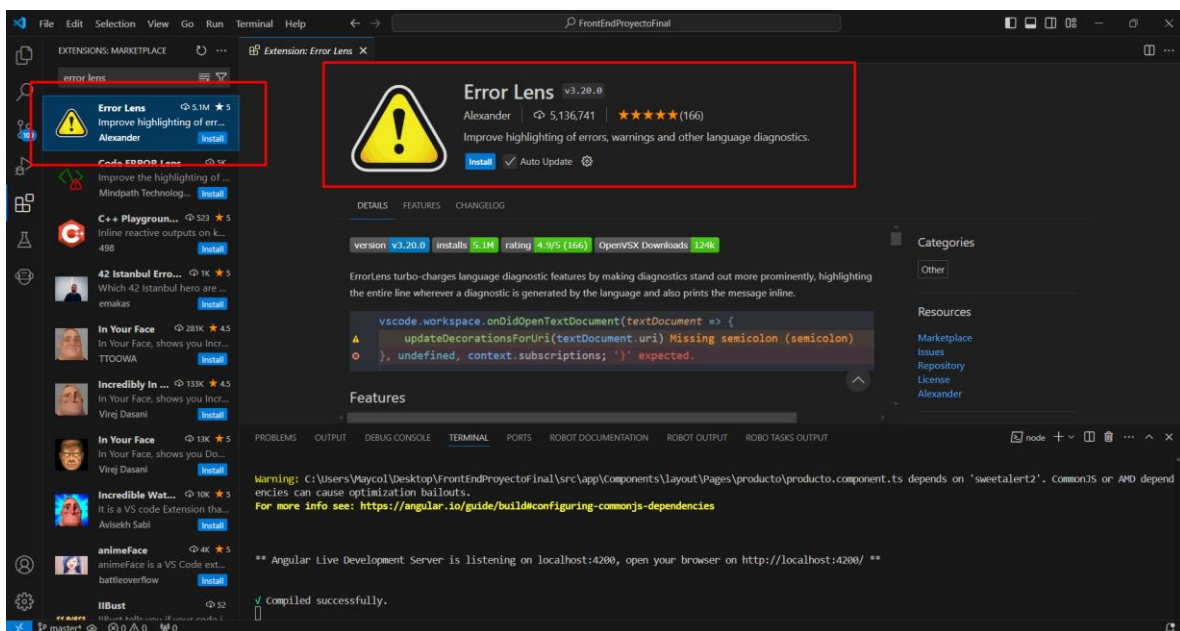
## **Introducción**

la seguridad se ha convertido en un aspecto fundamental para garantizar la protección de datos y la integridad de los sistemas. Este documento tiene como objetivo evaluar y documentar distintas herramientas de pruebas de seguridad utilizadas en el proceso de aseguramiento de calidad de software. Estas pruebas incluyen desde la detección de errores en el código hasta el análisis de vulnerabilidades en las dependencias de código abierto. Las cuales permiten identificar posibles riesgos y proponer soluciones efectivas para mitigarlos. Este análisis busca aportar un panorama detallado sobre la funcionalidad y la efectividad de cada herramienta en la mejora de la seguridad de aplicaciones de software.

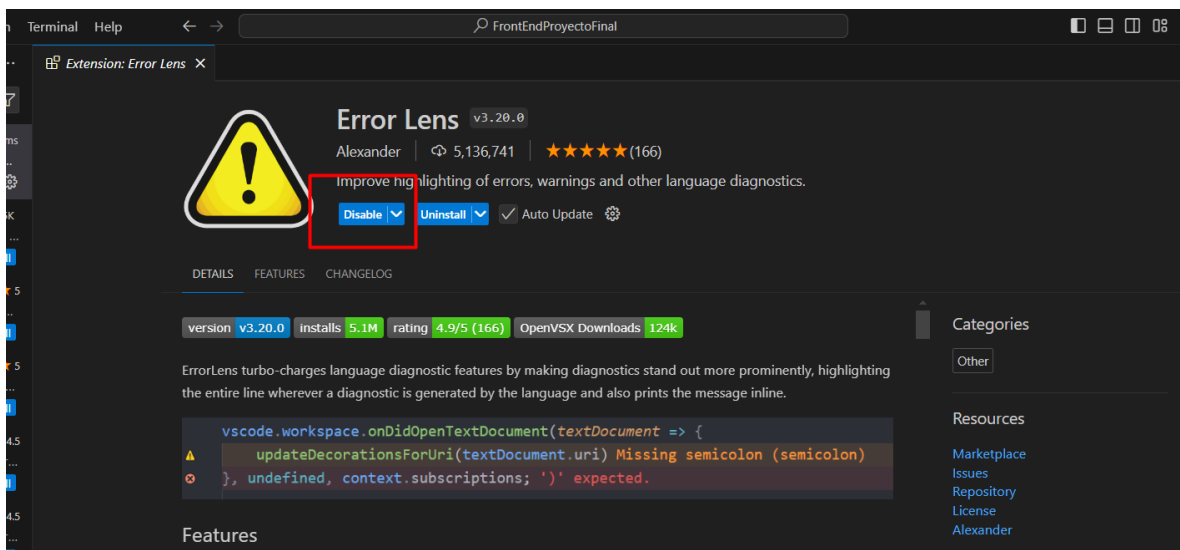
## Error Lens

Muestra un mensaje conciso y claro sobre el error directamente en la línea donde se produce, evitando la necesidad de abrir una ventana separada con los detalles.

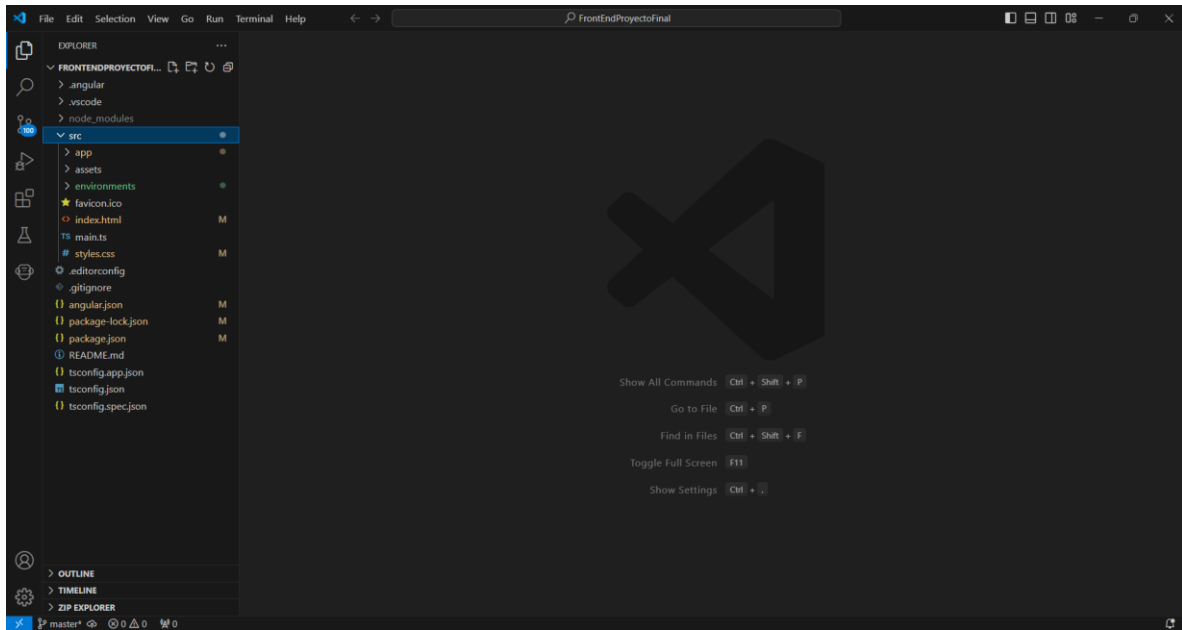
Procedemos a instalar la aplicación en el apartado de las extensiones.



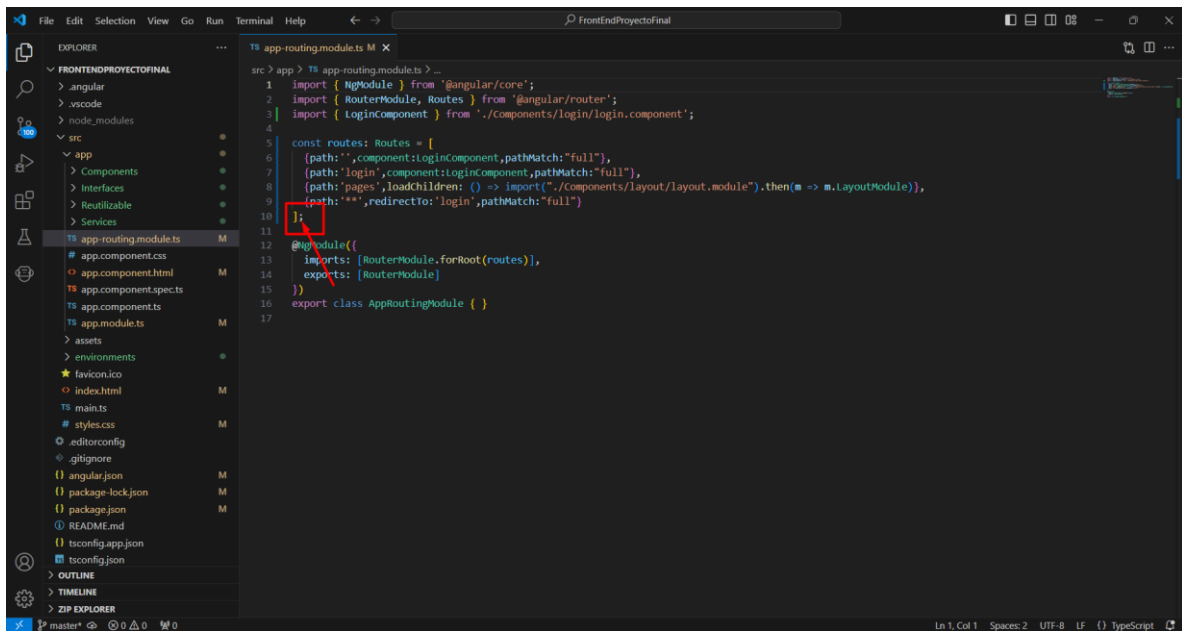
## Instalada



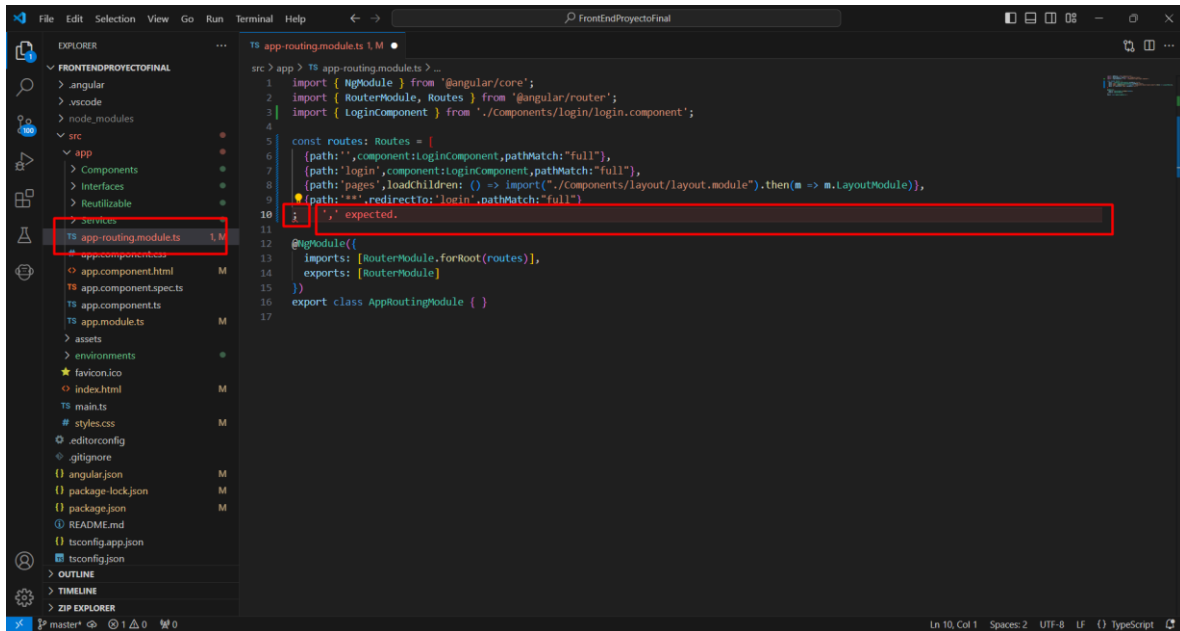
Procedemos a validar si funciona correctamente la instalación.



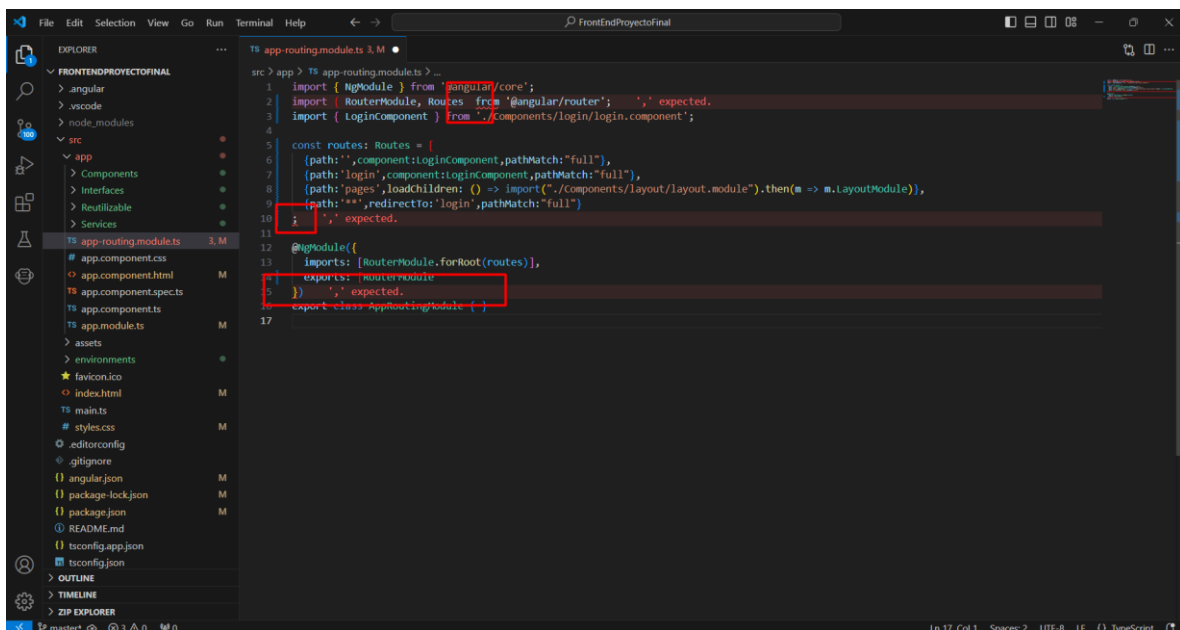
Se borra unos signos o lo que sea, para validar si funciona correctamente.



Automáticamente la aplicación detecta el error en nuestra línea de código.

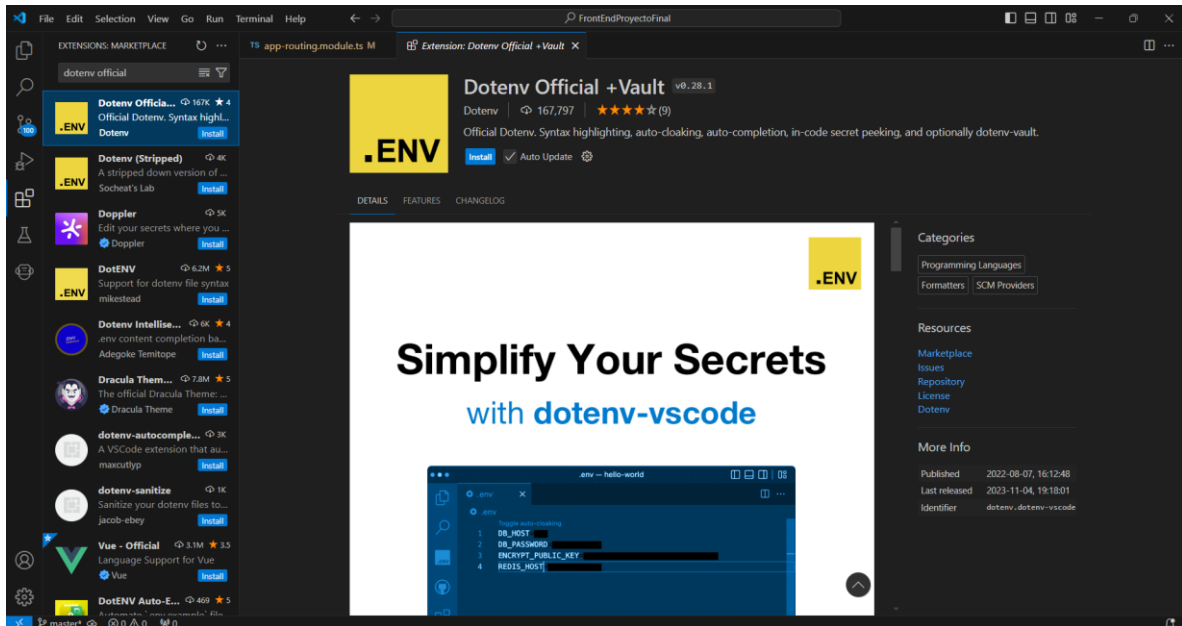


Nuevamente se hizo la prueba y efectivamente aparece el mensaje de “expected” indicando que tiene una falla de escritura.

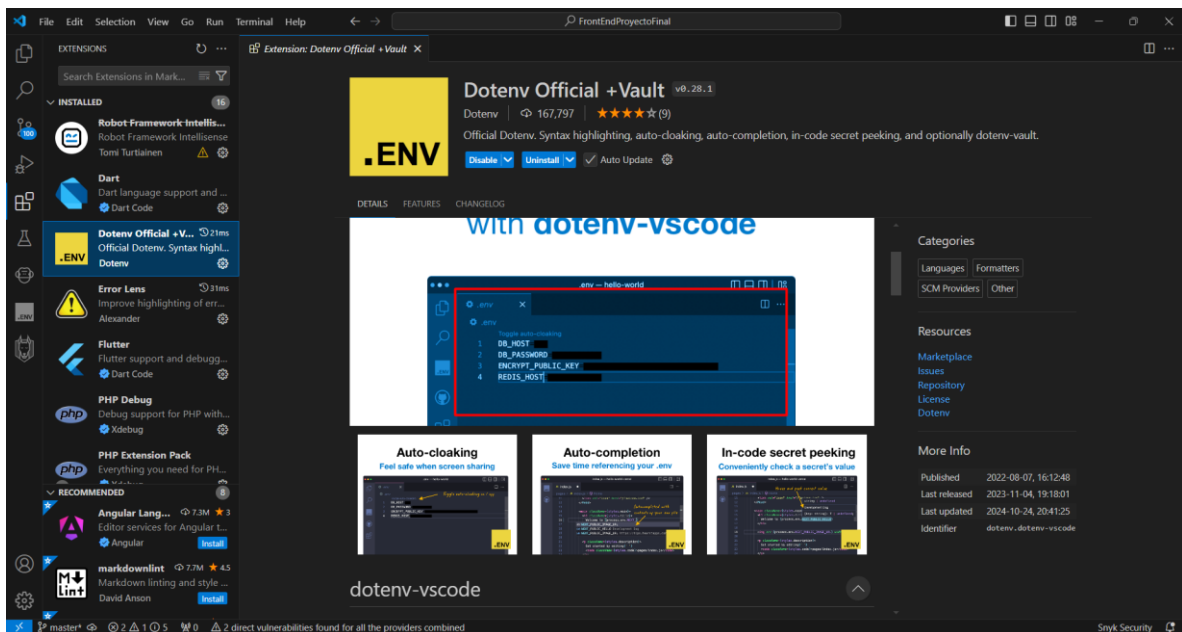


## Dotenv Official + Vault

Facilita la gestión de variables de entorno en tus proyectos.



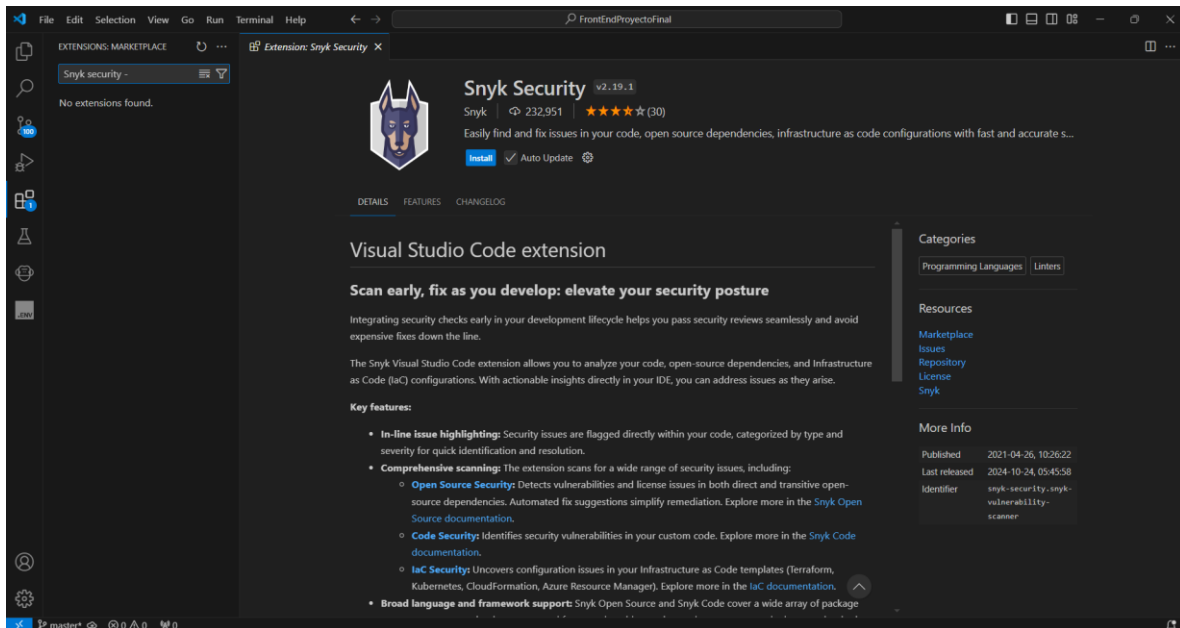
Cuando se instala la aplicación Dotenv, automáticamente se negrea las palabras importantes así como los archivos que terminan .env ya que para son esenciales los archivos para guardar información importante.



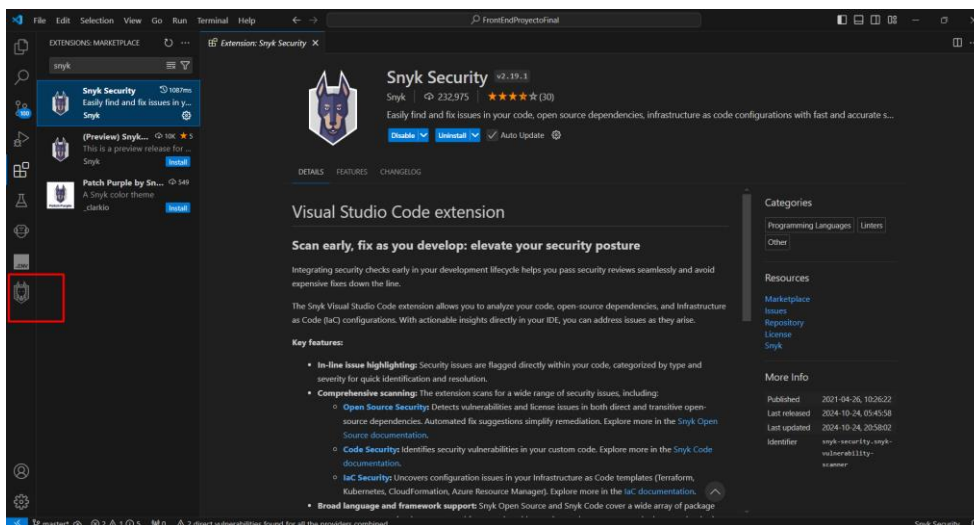
## Snyk Security – Code

es una herramienta muy útil para los desarrolladores que buscan mantener la seguridad de sus aplicaciones. Su principal función es detectar vulnerabilidades en tu código y en las dependencias de código abierto que se utiliza.

Se procede a instalar la aplicación Snyk Security

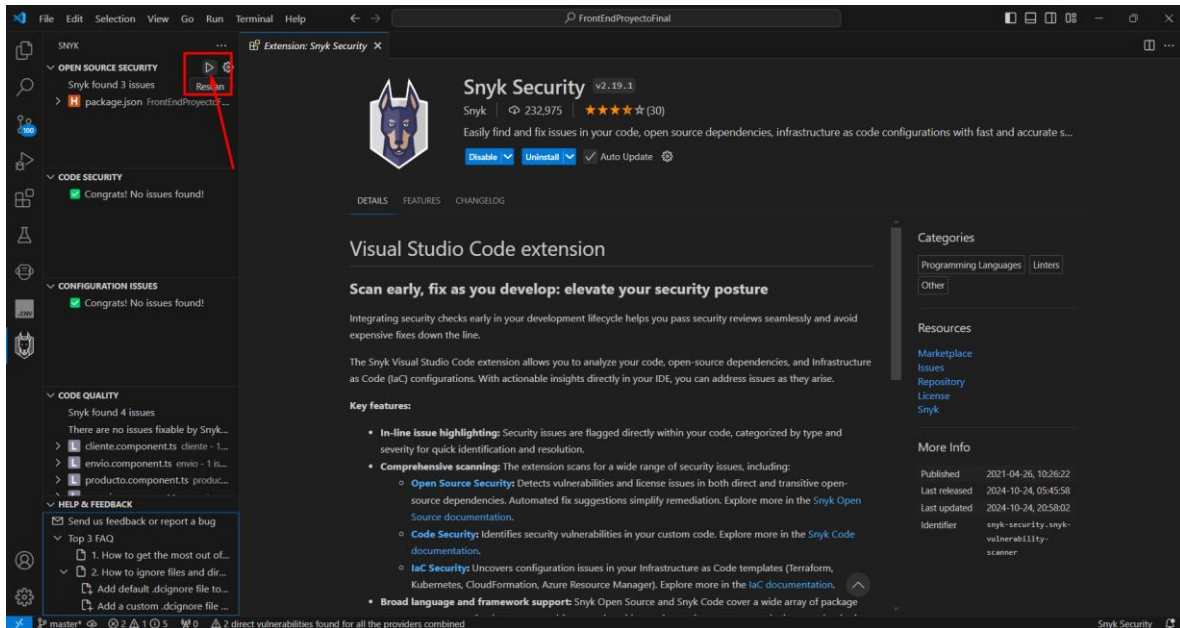


Instalada.

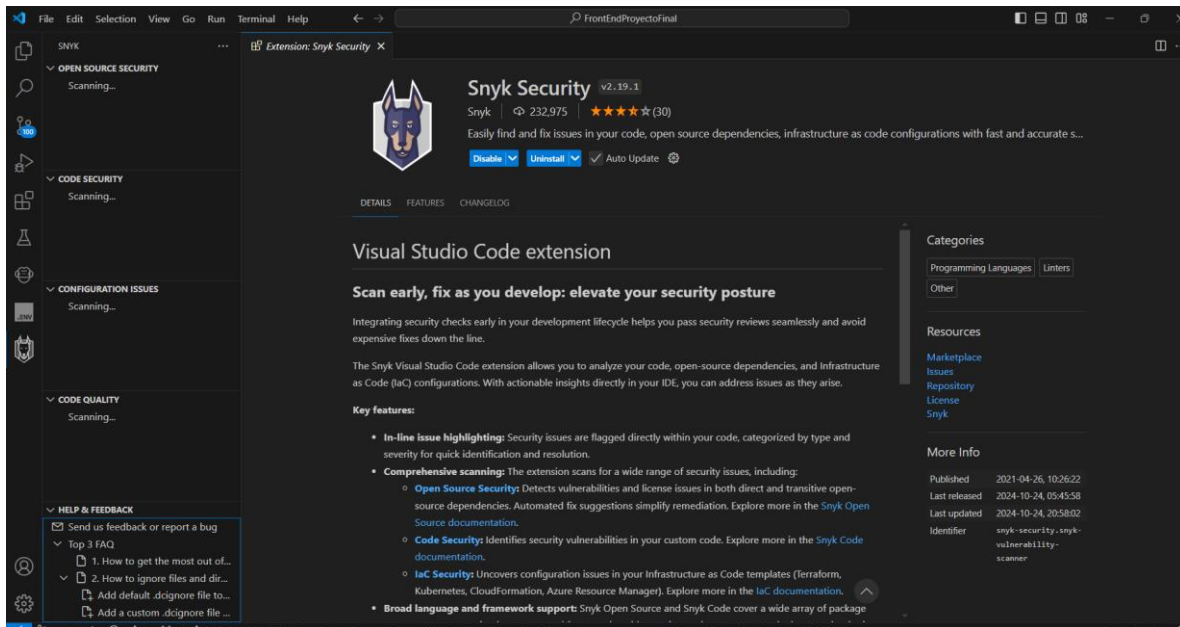




Presionar para iniciar con el escaneo de la aplicación.



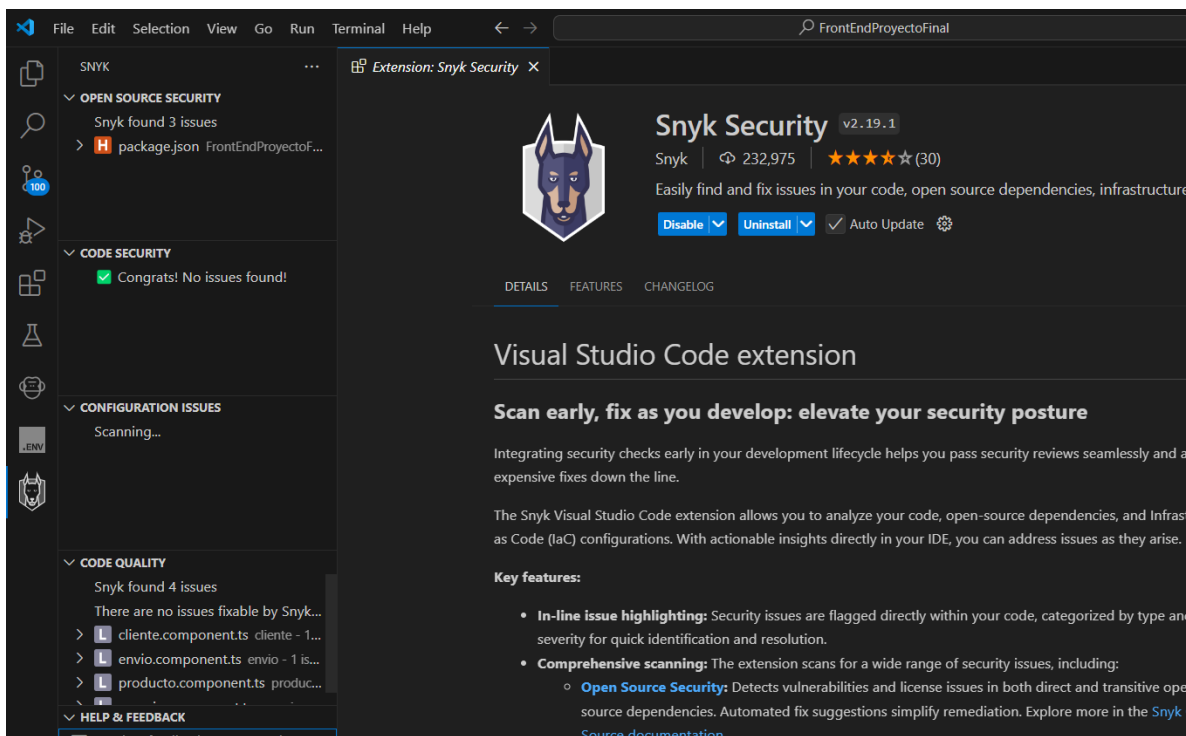
Empieza a realizar el escaneo del código.



Se analizan cuatro opciones:

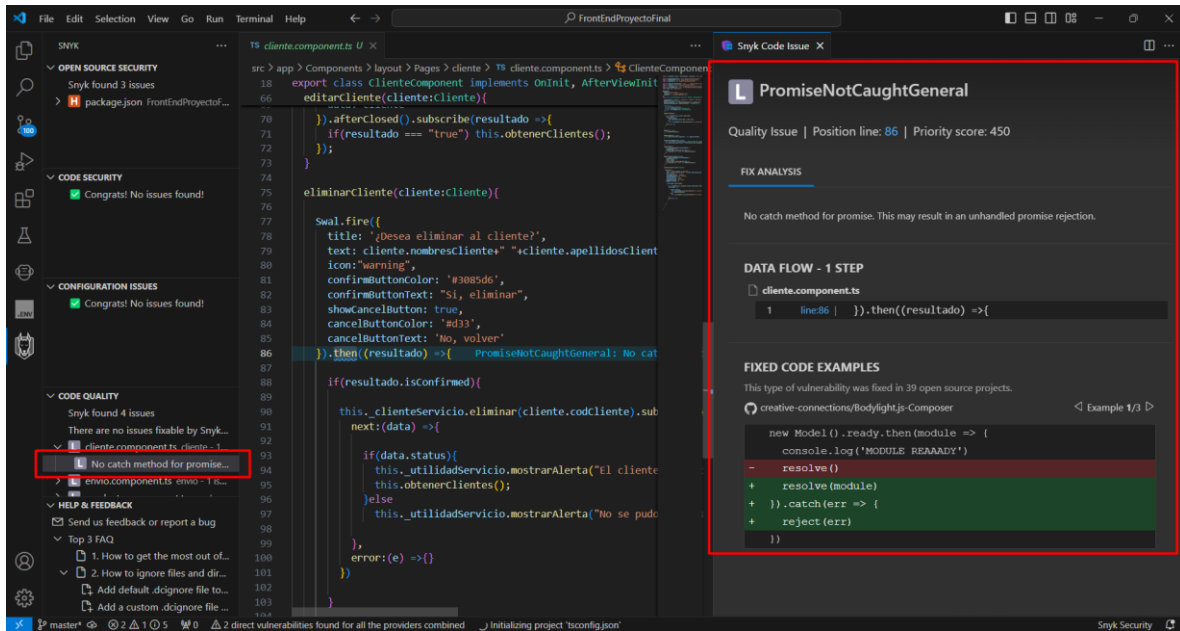
- OPEN SOURC SECURITY
- CODE SECURITY
- CONFIGURATION ISSUES
- CODE QUALITY

Dentro de estas se visualiza vulnerabilidades en el código, el cual también hace el análisis hacia nuestro backed.

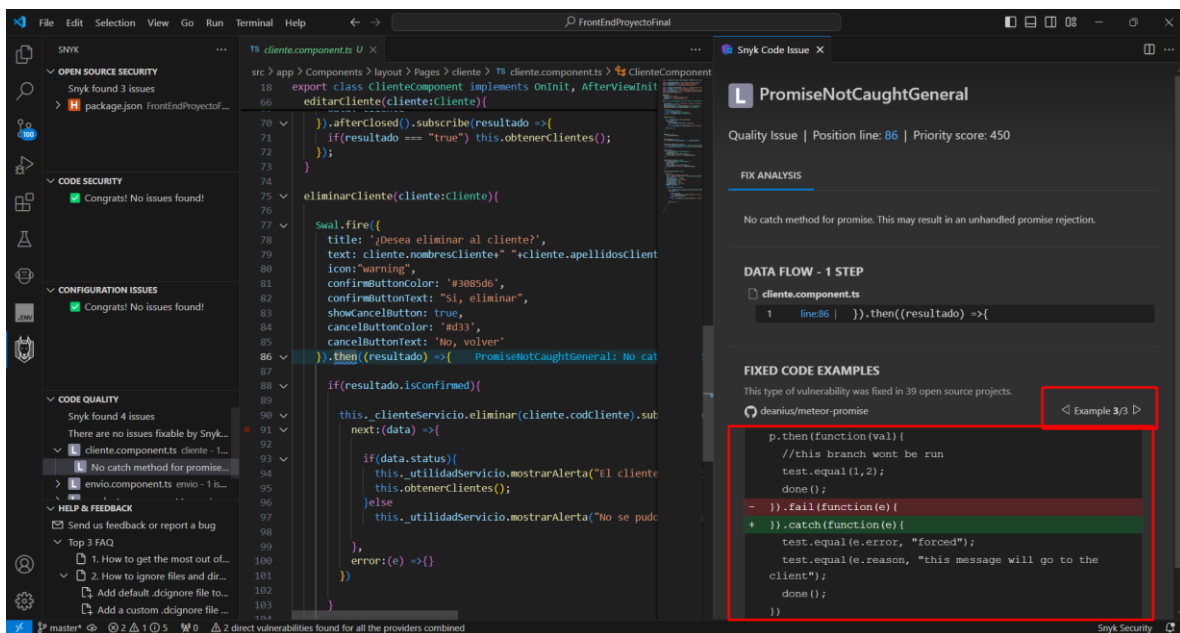


En el análisis se encontraron vulnerabilidades en dos opciones:

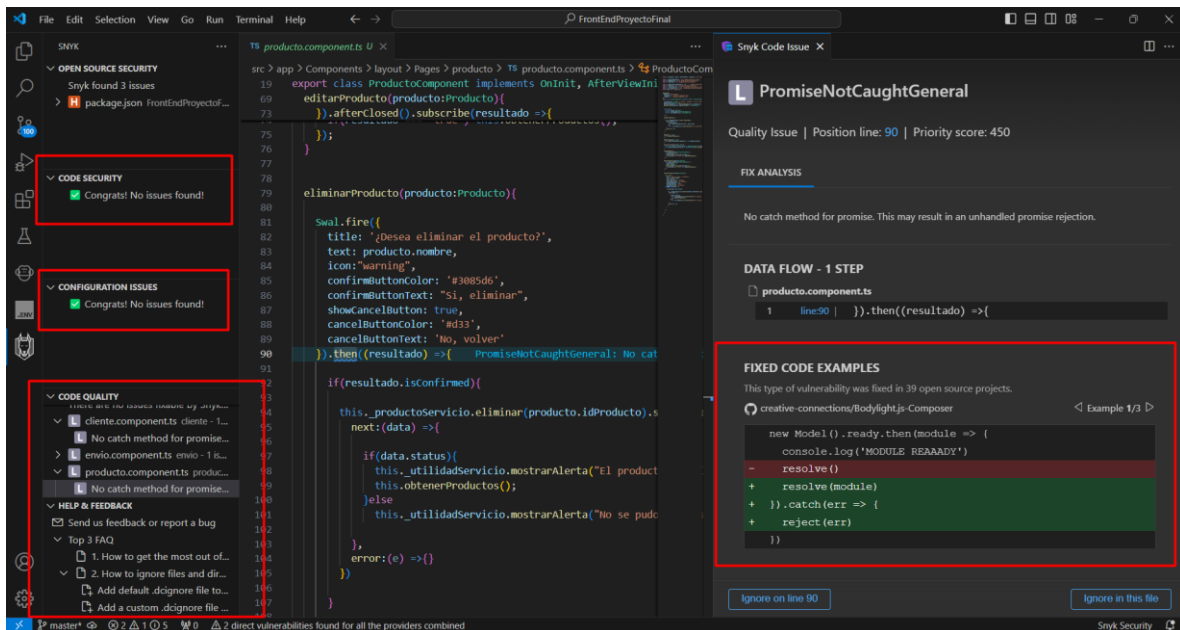
- OPEN SOURCE SECURITY
- CODE QUALITY



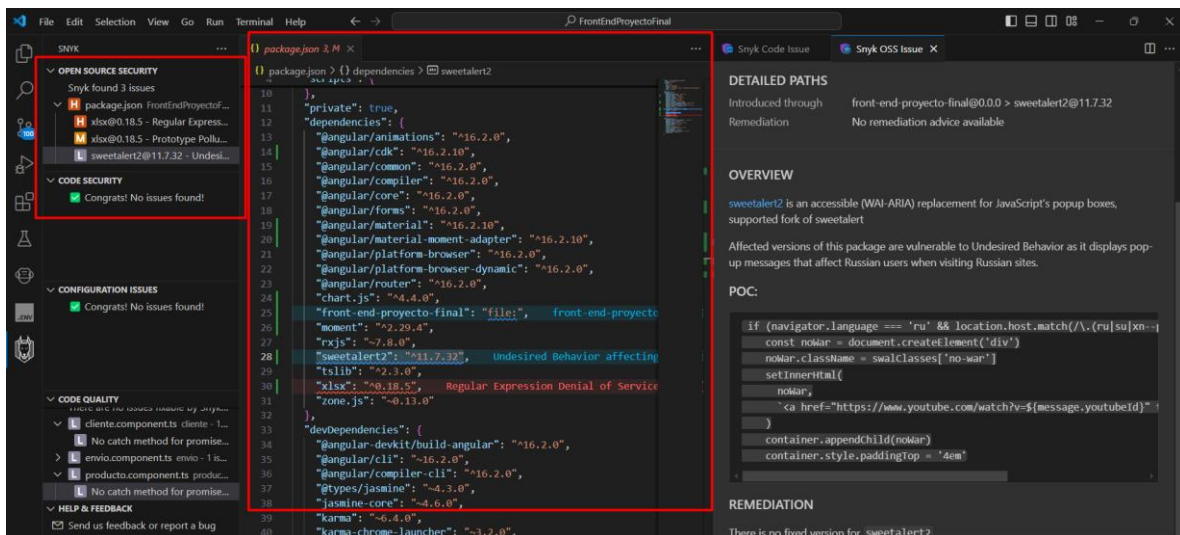
## Vulnerabilidad en CODE QUALITY



Sin vulnerabilidades en: CODE SECURITY Y CONFIGURATION ISSUES.

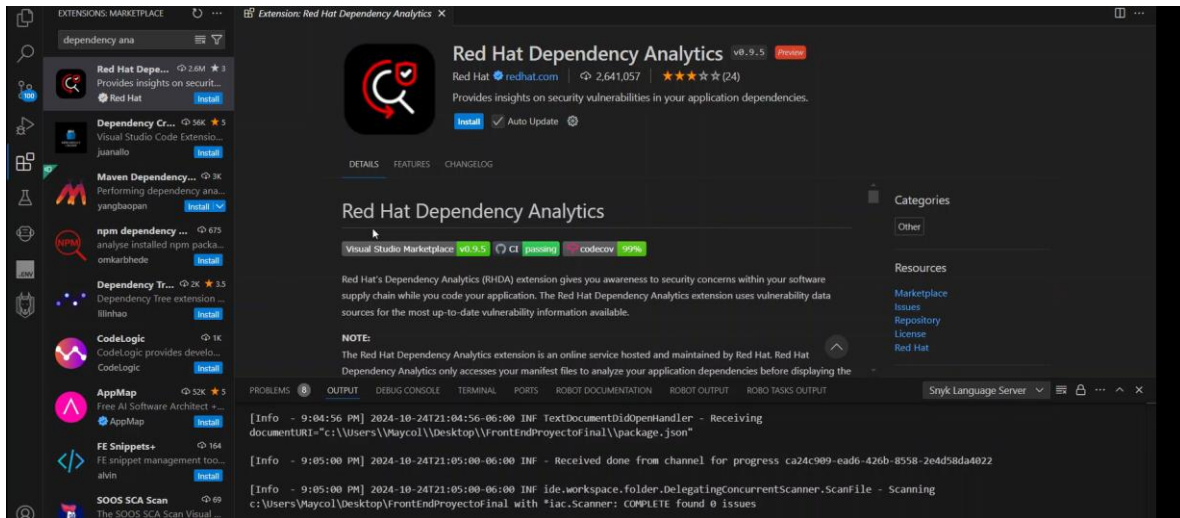


Vulnerabilidades en OPEN SOURCE SECURITY.

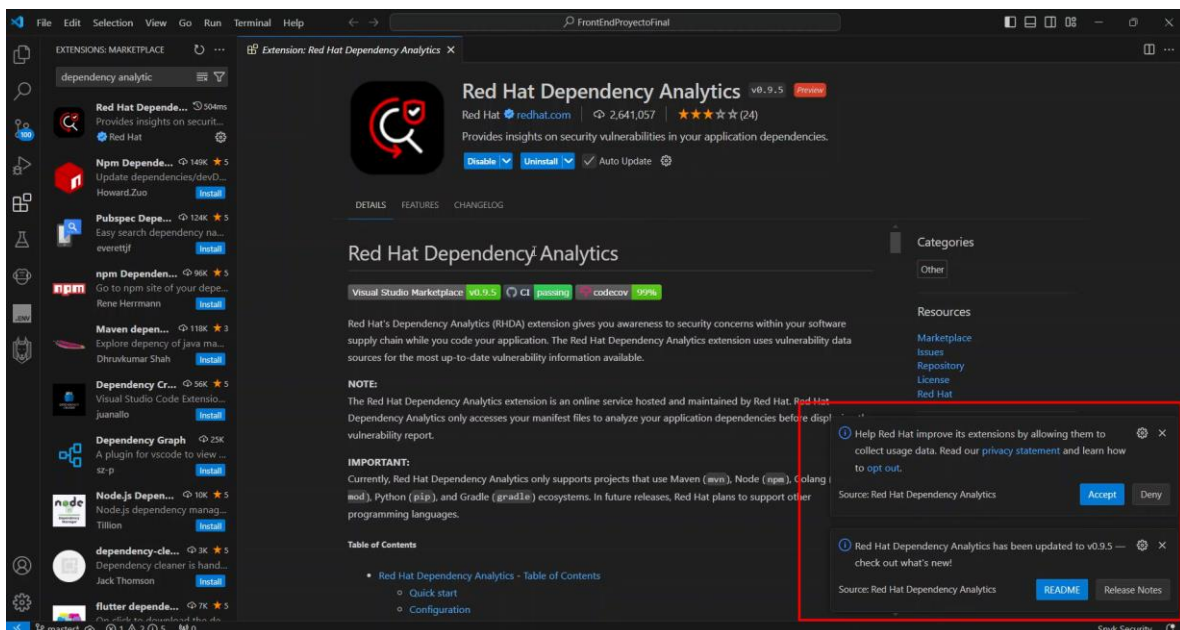


## Red Hat Dependency Analytics

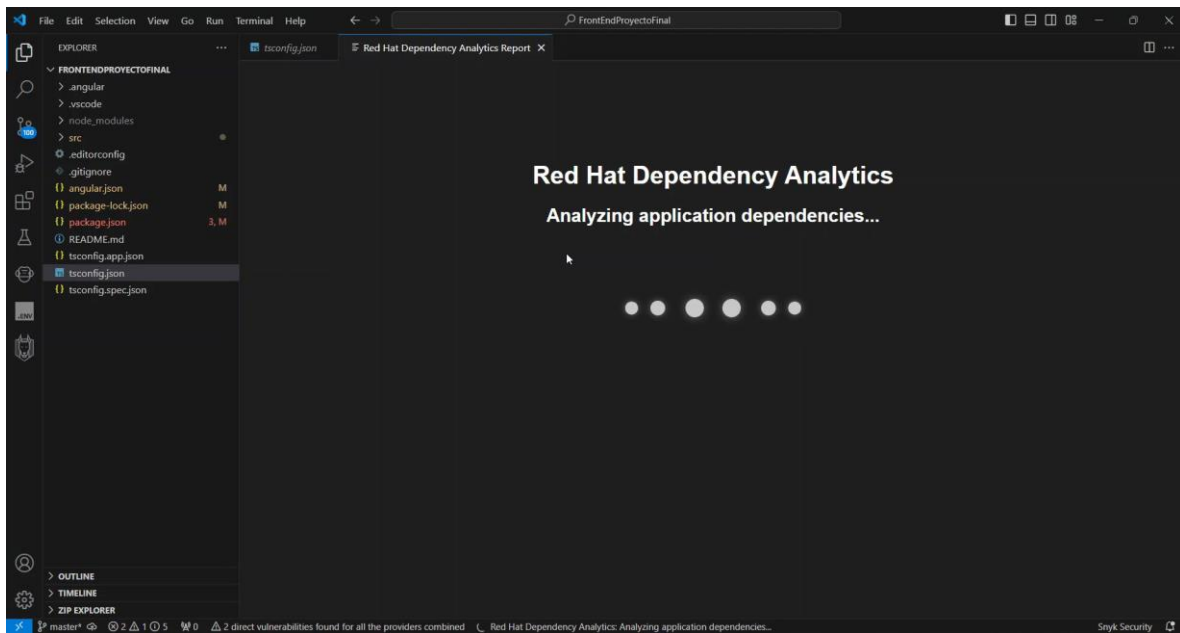
Es una herramienta valiosa para desarrolladores que utilizan tecnologías Java y Maven. Su principal función es analizar las dependencias de un proyecto y detectar posibles vulnerabilidades de seguridad.



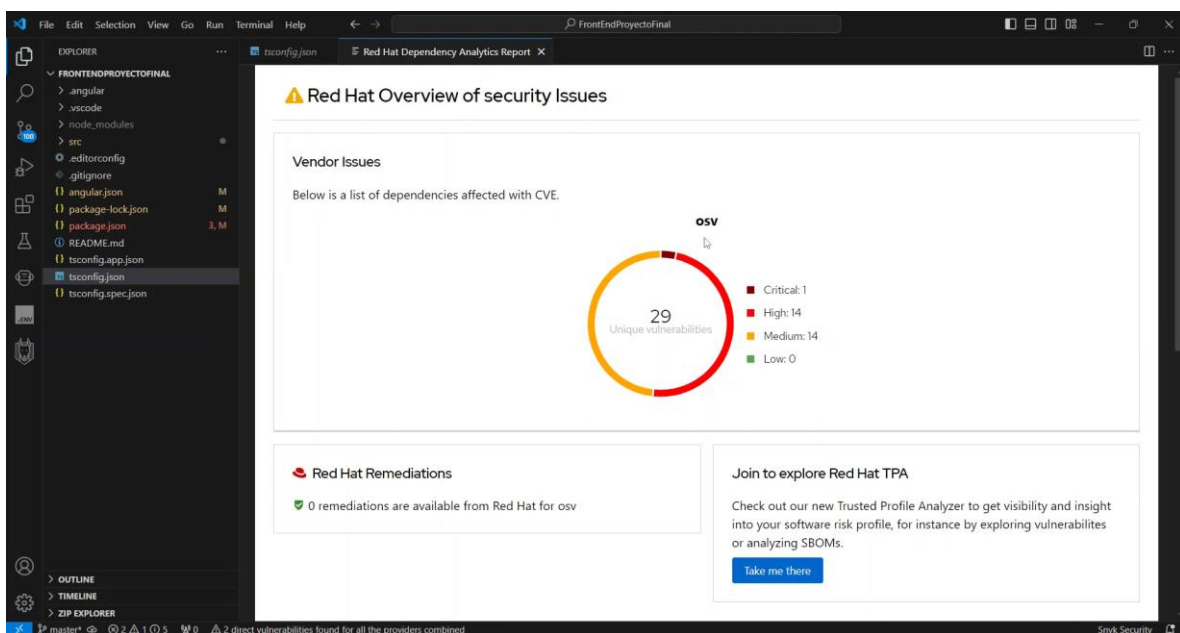
Instalado, presionar aceptar para que se pueda agregar.



Cuando termine la instalación, comenzara a realizar el escaneo.



Esta aplicación tiene como base a: “Snyk Security – Code” por lo que es mejor, en el reporte se logra apreciar que ha encontrado 29 vulnerabilidades, las cuales ha encontrado del backed y frontend, siendo más del frontend que del backed.





Vulnerabilidades encontradas en el código.

The screenshot shows the Red Hat Dependency Analytics Report in VS Code. The report is titled "Red Hat Dependency Analytics Report" and is for the project "FrontEndProyectoFinal". It displays a table of dependencies and their vulnerabilities.

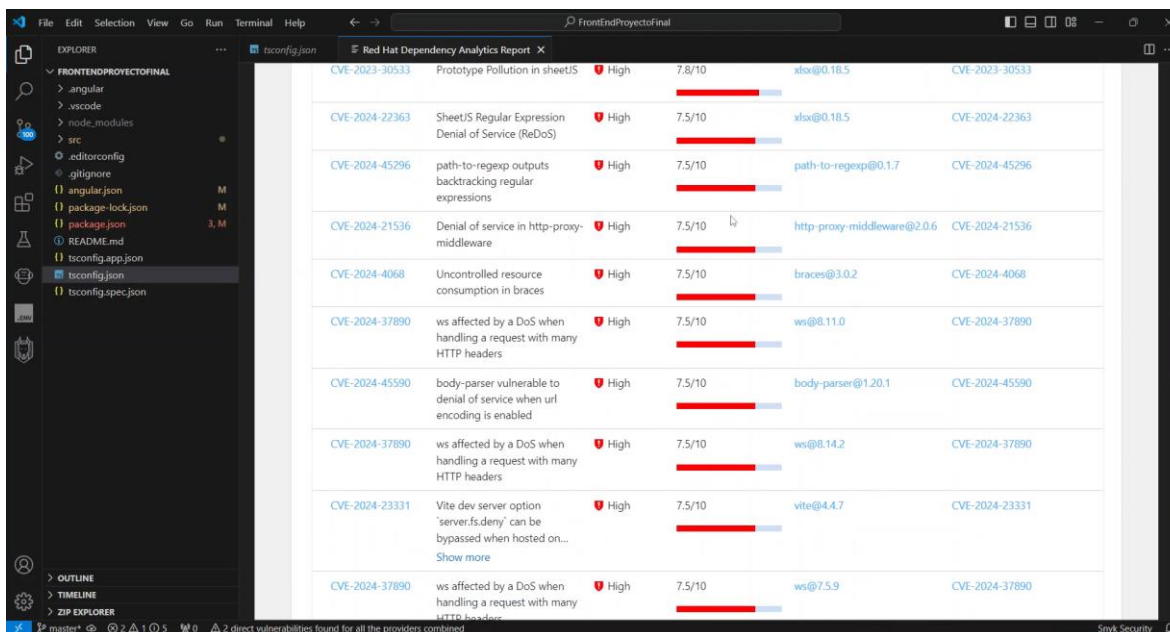
Dependency Name	Current Version	Direct Vulnerabilities	Transitive Vulnerabilities	Remediation available...
angular/cli	16.2.8	0	4	Yes
front-end-proyecto-final	0.0.0	0	29	Yes
xlsx	0.18.5	2	0	Yes
angular/compiler-cli	16.2.10	0	1	Yes
karma-jasmine-html-reporter	2.1.0	0	5	Yes
angular-devkit/build-angular	16.2.8	0	24	Yes
karma-jasmine	5.1.0	0	5	Yes
karma	6.4.2	0	5	Yes

Si se presiona el modulo, desplegara las vulnerabilidades que ha encontrado.

The screenshot shows the Red Hat Dependency Analytics Report in VS Code. The report is titled "Red Hat Dependency Analytics Report" and is for the project "FrontEndProyectoFinal". It displays a table of dependencies and their vulnerabilities.

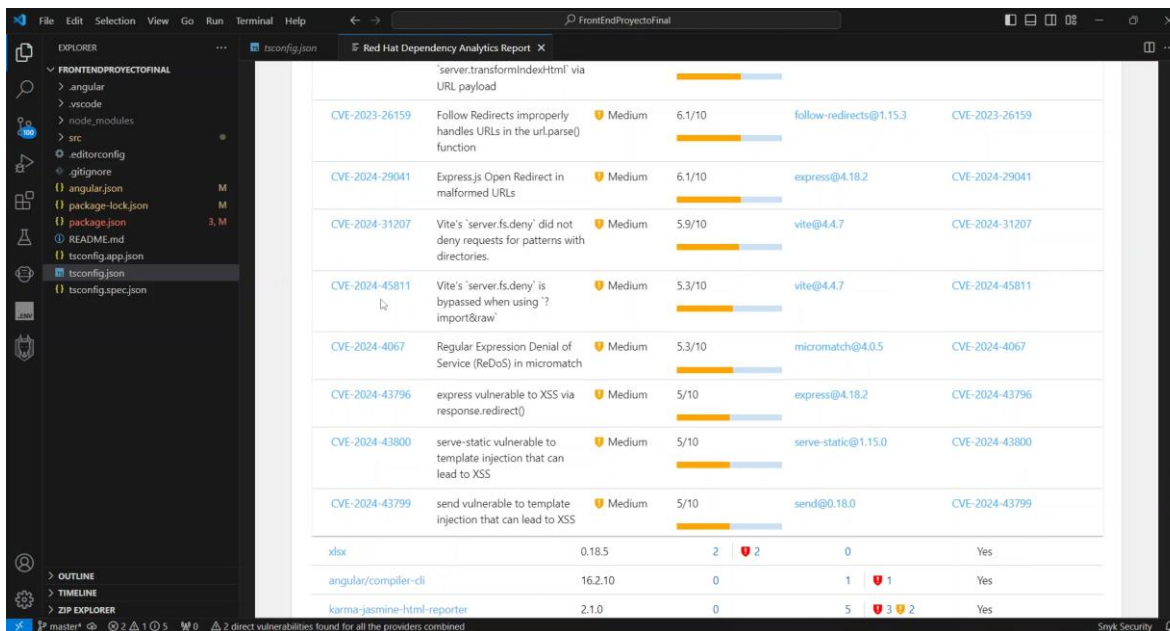
Dependency Name	Current Version	Direct Vulnerabilities	Transitive Vulnerabilities	Remediation available...
angular/cli	16.2.8	0	4	Yes
front-end-proyecto-final	0.0.0	0	29	Yes
xlsx	0.18.5	2	0	Yes
angular/compiler-cli	16.2.10	0	1	Yes
karma-jasmine-html-reporter	2.1.0	0	5	Yes
angular-devkit/build-angular	16.2.8	0	24	Yes
karma-jasmine	5.1.0	0	5	Yes
karma	6.4.2	0	5	Yes

## Vulnerabilidades del frontend, en termino alto.



CVE	Description	Severity	Score	Package	Version
CVE-2023-30533	Prototype Pollution in sheetJS	High	7.8/10	xlsx@0.18.5	CVE-2023-30533
CVE-2024-22363	SheetJS Regular Expression Denial of Service (ReDoS)	High	7.5/10	xlsx@0.18.5	CVE-2024-22363
CVE-2024-45296	path-to-regexp outputs backtracking regular expressions	High	7.5/10	path-to-regexp@0.1.7	CVE-2024-45296
CVE-2024-21536	Denial of service in http-proxy-middleware	High	7.5/10	http-proxy-middleware@2.0.6	CVE-2024-21536
CVE-2024-4068	Uncontrolled resource consumption in braces	High	7.5/10	braces@3.0.2	CVE-2024-4068
CVE-2024-37890	ws affected by a DoS when handling a request with many HTTP headers	High	7.5/10	ws@8.11.0	CVE-2024-37890
CVE-2024-45590	body-parser vulnerable to denial of service when url encoding is enabled	High	7.5/10	body-parser@1.20.1	CVE-2024-45590
CVE-2024-37890	ws affected by a DoS when handling a request with many HTTP headers	High	7.5/10	ws@8.14.2	CVE-2024-37890
CVE-2024-23331	Vite dev server option 'server.fs.deny' can be bypassed when hosted on...	High	7.5/10	vite@4.4.7	CVE-2024-23331
CVE-2024-37890	ws affected by a DoS when handling a request with many HTTP headers	High	7.5/10	ws@7.5.9	CVE-2024-37890

## Vulnerabilidades del backed, en término medio.



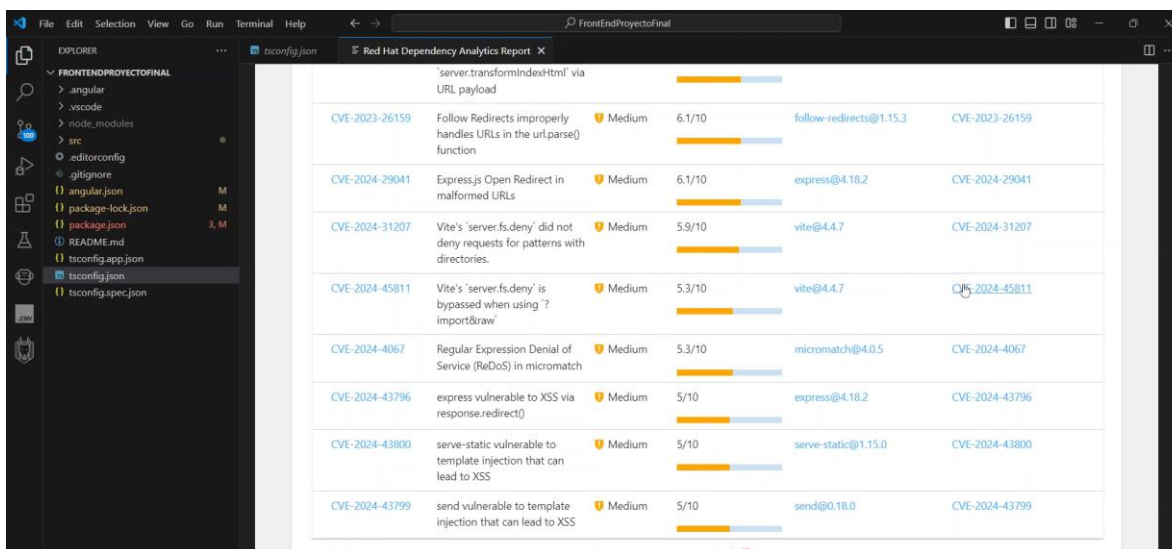
CVE	Description	Severity	Score	Package	Version
CVE-2023-26159	Follow Redirects improperly handles URLs in the url.parse() function	Medium	6.1/10	follow-redirects@1.15.3	CVE-2023-26159
CVE-2024-29041	Express.js Open Redirect in malformed URLs	Medium	6.1/10	express@4.18.2	CVE-2024-29041
CVE-2024-31207	Vite's 'server.fs.deny' did not deny requests for patterns with directories.	Medium	5.9/10	vite@4.4.7	CVE-2024-31207
CVE-2024-45811	Vite's 'server.fs.deny' is bypassed when using '? import&raw'	Medium	5.3/10	vite@4.4.7	CVE-2024-45811
CVE-2024-4067	Regular Expression Denial of Service (ReDoS) in micromatch	Medium	5.3/10	micromatch@4.0.5	CVE-2024-4067
CVE-2024-43796	express vulnerable to XSS via response.redirect()	Medium	5/10	express@4.18.2	CVE-2024-43796
CVE-2024-43800	serve-static vulnerable to template injection that can lead to XSS	Medium	5/10	serve-static@1.15.0	CVE-2024-43800
CVE-2024-43799	send vulnerable to template injection that can lead to XSS	Medium	5/10	send@0.18.0	CVE-2024-43799

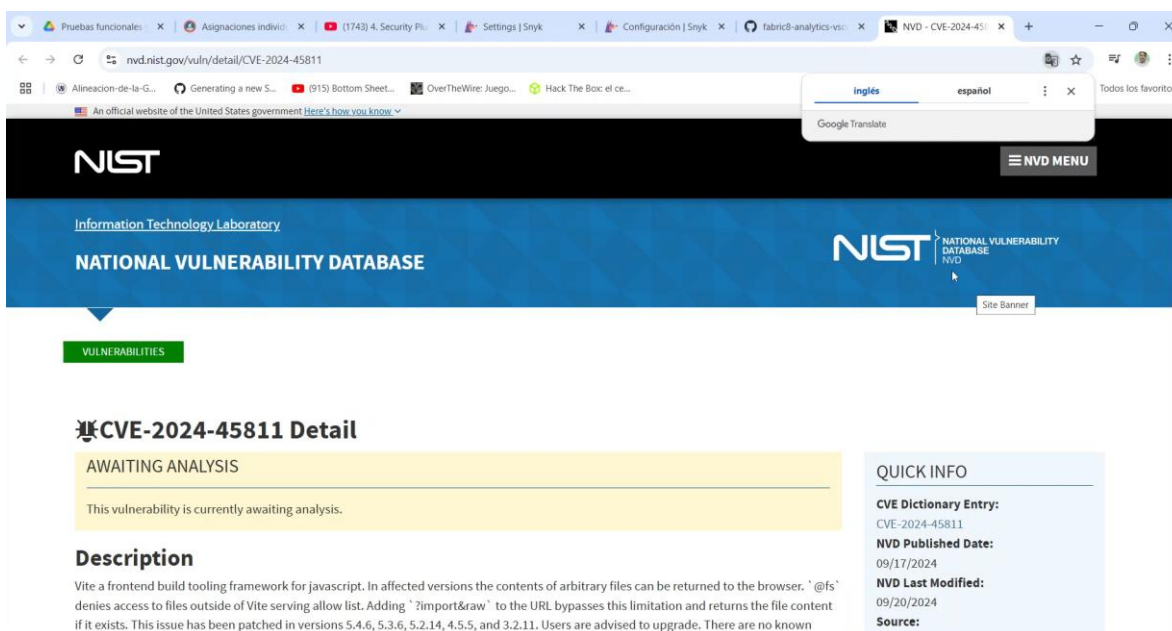
Package	Version	Score	Severity	Count	Yes
xlsx	0.18.5	2	High	2	0
angular/compiler-cli	16.2.10	0	Low	1	1
karma-jasmine-html-reporter	2.1.0	0	Low	5	3



Al presionar la opción de lado, izquierdo nos lleva a la solución y como se debe de proceder hacia la vulnerabilidad encontrada.



Página que nos da la solución a nuestra vulnerabilidad.



## Pasos para seguir y corregir.

← → ↻ nvd.nist.gov/vuln/detail/CVE-2024-45811

Alineacion-de-la-G... Generating a new S... (915) Bottom Sheet... OverTheWire: Juego... Hack The Box: el ce... workarounds for this vulnerability.



inglés español Google Translate Todos los favoritos

### Metrics

CVSS Version 4.0 CVSS Version 3.x CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

**CVSS 3.x Severity and Vector Strings:**

 <b>NIST:</b> NVD	<b>Base Score:</b> N/A	NVD assessment not yet provided.
 <b>CNA:</b> GitHub, Inc.	<b>Base Score:</b> 4.8 MEDIUM	<b>Vector:</b> CVSS:3.1/AV:A/AC:H/PR:N/UI:R/S:U/C:H/I:N/A:N

### References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to [nvd@nist.gov](mailto:nvd@nist.gov).

Hypertlink	Resource
<a href="https://github.com/vitejs/vite/commit/6820bb3b9a54334f3268fc5ee1e967d2e1c0db34">https://github.com/vitejs/vite/commit/6820bb3b9a54334f3268fc5ee1e967d2e1c0db34</a>	
<a href="https://github.com/vitejs/vite/security/advisories/GHSA-9cwx-2883-4wfx">https://github.com/vitejs/vite/security/advisories/GHSA-9cwx-2883-4wfx</a>	

### Weakness Enumeration

CWE-ID	CWE Name	Source
--------	----------	--------

## Análisis de Red Hat.

File Edit Selection View Go Run Terminal Help

FrontEndProjectoFinal

tsconfig.json Red Hat Dependency Analytics Report

### Red Hat Overview of security Issues

**Vendor Issues**

Below is a list of dependencies affected with CVE.

**OSV**

29 Unique vulnerabilities

- Critical: 1
- High: 14
- Medium: 14
- Low: 0

**Red Hat Remediations**

0 remediations are available from Red Hat for osv

**Join to explore Red Hat TPA**

Check out our new Trusted Profile Analyzer to get visibility and insight into your software risk profile, for instance by exploring vulnerabilities or analyzing SBOMs.

[Take me there](#)

2 direct vulnerabilities found for all the providers combined

Snyk Security

## **Conclusión**

En conclusión, el uso de herramientas de pruebas de seguridad en el desarrollo de software es esencial para fortalecer la protección y la calidad de las aplicaciones. Las herramientas evaluadas en este documento, como Error Lens y Snyk Security – Code, han demostrado ser eficaces en la identificación de vulnerabilidades y en la mejora de la seguridad en el código y sus dependencias. Cada herramienta aporta un enfoque único que, en conjunto, permite a los desarrolladores abordar la seguridad desde múltiples perspectivas. Al integrar estas herramientas en el flujo de trabajo de desarrollo, se promueve una cultura de seguridad que contribuye a reducir riesgos y a construir aplicaciones más confiables y seguras.