

INFORME TECNICO
APLICACIÓN DE RESERVAS DE PLANES TURÍSTICOS

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
ESPECIALIZACIÓN EN CIBERSEGURIDAD
SEG EN ENTORNOS CLOUD Y DEVOPS
JHON ALEXANDER PINTO
HERIBERTO TIRADO PINZÓN
JOHANY CASTRO
JUAN PABLO RAMÍREZ EDWIN
EDWIN MAURICIO HERRERA



Contenido

RESUMEN EJECUTIVO	4
OBJETIVOS DEL PROYECTO:	4
METODOLOGÍA Y PRÁCTICAS:	5
• Integración Continua y Entrega Continua (CI/CD)	5
• Prácticas de Seguridad Robustas	5
• Control de Versiones:	5
• Documentación Clara y Concisa	6
• Contenerización	6
DIAGRAMA DE DESPLIEGUE	6
CASOS DE USO	7
1. Registro de Usuario:.....	7
2. Inicio de sesión:	7
3. Consulta de Planes:.....	8
4. Ver detalle del Plan:.....	8
5. Realizar Reserva:	8
6. Ver mis Reservas:	9
HISTORIAS DE USUARIO	9
1. Registro de Usuario:.....	9
2. Inicio de Sesión:	9
3. Consulta de Planes:.....	9
4. Ver Detalle del Plan:	10
5. Realizar Reserva:	10
6. Ver Mis Reservas (Opcional/Mejora):	10
SEGURIDAD	10

¿Qué es Snyk?.....	10
¿Cómo funciona Snyk?	10
GITHUB	11
DEPENDABOT	12
VULNERABILIDAD CRITICA	12
GRAVEDAD.....	13
VULNERABILIDAD EN SYLIUS (CVE-2024-34349).....	14
VULNERABILIDAD BAJA	14
IMPACTO	14
PARCHES	14
SOLUCIONES ALTERNATIVAS.....	14
GitHub Code Secutity	15
Amazon Code Review	16

TABLA DE IMÁGENES

Ilustración 1 Diagrama.....	6
Ilustración 2 escaneo código con SNYK	11
Ilustración 3 Código GitHub	12
Ilustración 4 Análisis Vulnerabilidad github.....	12

RESUMEN EJECUTIVO

Este informe presenta el estado actual y la visión general del proyecto TourisPlan, una aplicación web desarrollada con Next.js y JavaScript para la reserva de planes turísticos en Colombia. El objetivo principal es proporcionar una plataforma intuitiva y segura donde los usuarios puedan registrarse, autenticarse, explorar las diferentes ofertas turísticas, y realizar reservas de manera eficiente. El proyecto se está llevando a cabo bajo una metodología ágil, con un enfoque en la entrega continua a través de un pipeline de CI/CD, la implementación de prácticas de seguridad robustas y la generación de documentación técnica relevante.

OBJETIVOS DEL PROYECTO:

El objetivo primordial del proyecto TourisPlan es desarrollar e implementar una aplicación web completa que permita a los usuarios:

- Registrarse y autenticarse de forma segura.
- Consultar una variedad de planes turísticos disponibles en Colombia, obtenidos a través de la integración con la API externa "api-Colombia".
- Visualizar los planes turísticos de manera atractiva y organizada mediante tarjetas informativas.

- Acceder a una vista detallada de cada plan turístico, incluyendo información relevante y multimedia.
- Realizar reservas de los planes turísticos de su interés, una vez autenticados.
- Almacenar de forma segura los datos de usuario y las reservas en una base de datos externa.
- Gestionar el código fuente de manera eficiente a través de un repositorio en GitHub.
- Automatizar el proceso de construcción, prueba e implementación de la aplicación mediante un pipeline de CI/CD con GitHub Actions.
- Integrar análisis y escaneo de seguridad para identificar y mitigar posibles vulnerabilidades.
- Generar documentación técnica esencial para la administración y el despliegue de la aplicación (Manual de Administrador, Diagrama de Despliegue).
- Facilitar la portabilidad y escalabilidad de la aplicación mediante su contenerización con Docker.

METODOLOGÍA Y PRÁCTICAS:

El proyecto TourisPlan se desarrollará bajo un marco de trabajo ágil, lo que permitirá una mayor flexibilidad y adaptabilidad a los cambios durante el ciclo de vida del desarrollo. Se priorizará la colaboración continua, la entrega iterativa y la retroalimentación temprana.

Además, se implementarán las siguientes prácticas clave:

- **Integración Continua y Entrega Continua (CI/CD):** Automatización de los procesos de construcción, prueba e implementación para garantizar entregas frecuentes y confiables.
- **Prácticas de Seguridad Robustas:** Incorporación de medidas de seguridad en todas las etapas del desarrollo, incluyendo el análisis de vulnerabilidades con CVSS 4.0 para identificar y mitigar riesgos potenciales.
- **Control de Versiones:** Utilización de GitHub para la gestión del código fuente, facilitando la colaboración y el seguimiento de los cambios.

- **Documentación Clara y Concisa:** Generación de documentación técnica relevante para facilitar la comprensión, el mantenimiento y la administración de la aplicación.
- **Contenerización:** Uso de Docker para empaquetar la aplicación, simplificando el despliegue y garantizando la consistencia en diferentes entornos.

DIAGRAMA DE DESPLIEGUE

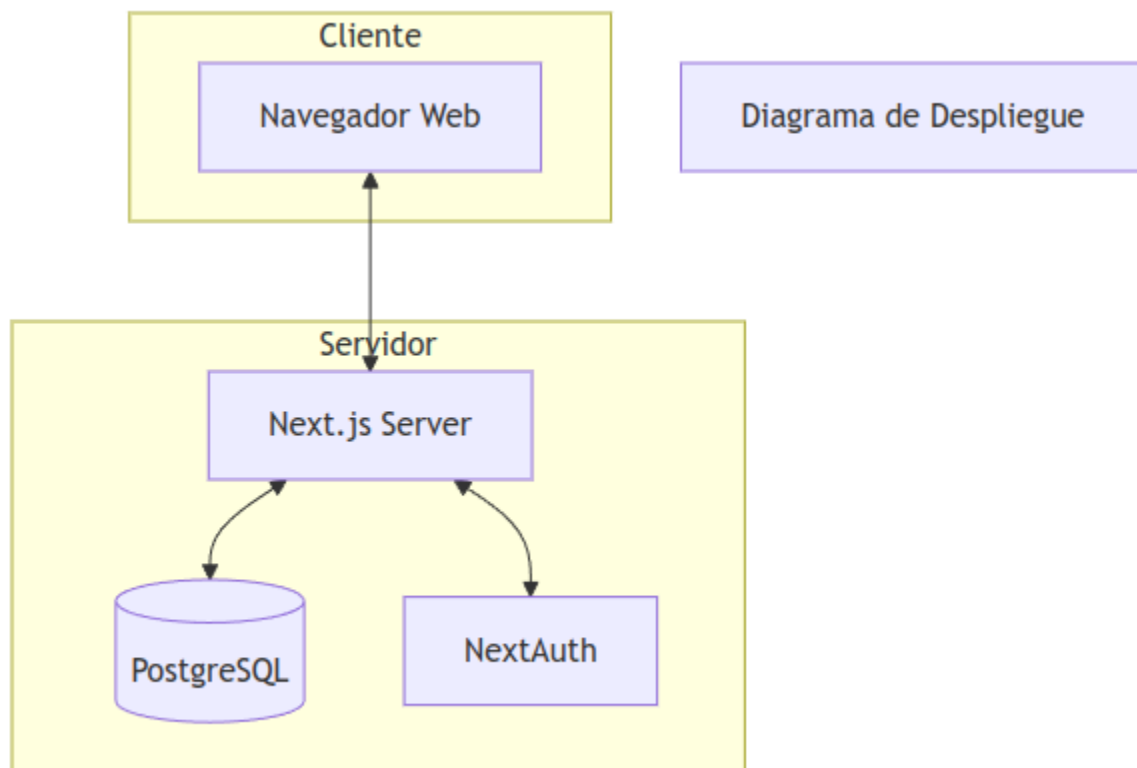


Ilustración 1 Diagrama

CASOS DE USO

1. Registro de Usuario:

Campo	Descripción
Nombre	Registro de Usuario
Actor Principal	Visitante
Descripción	Permite al visitante registrarse en la aplicación con correo electrónico y contraseña.
Precondiciones	El visitante no debe estar registrado previamente.
Flujo Principal	1. El visitante accede al formulario de registro. 2. Ingresa correo y contraseña. 3. Envía el formulario. 4. El sistema valida y crea la cuenta.
Flujo Alternativo	3a. Si el correo ya está registrado, el sistema muestra un mensaje de error.
Postcondiciones	El usuario queda registrado y puede iniciar sesión.
Requisitos Especiales	Validación de correo y fuerza de la contraseña.

2. Inicio de sesión:

Campo	Descripción
Nombre	Inicio de Sesión
Actor Principal	Usuario Registrado
Descripción	Permite al usuario iniciar sesión con su correo y contraseña.
Precondiciones	El usuario debe estar registrado.
Flujo Principal	1. El usuario accede al formulario de inicio de sesión. 2. Ingresa sus credenciales. 3. El sistema verifica y autentica.
Flujo Alternativo	3a. Si las credenciales son incorrectas, el sistema muestra un mensaje de error.
Postcondiciones	El usuario accede a su cuenta.
Requisitos Especiales	Cifrado de contraseñas, protección contra ataques de fuerza bruta.

3. Consulta de Planes:

Campo	Descripción
Nombre	Consulta de Planes
Actor Principal	Usuario Autenticado
Descripción	Permite visualizar una lista de planes turísticos obtenidos de api-Colombia.
Precondiciones	El usuario debe haber iniciado sesión.
Flujo Principal	1. El usuario accede a la sección de planes. 2. El sistema consulta api-Colombia. 3. Muestra los planes en tarjetas.
Flujo Alternativo	2a. Si hay error al consultar la API, se muestra mensaje informativo.
Postcondiciones	Se presenta la lista de planes turísticos.
Requisitos Especiales	Conexión a la API y manejo de errores.

4. Ver detalle del Plan:

Campo	Descripción
Nombre	Ver Detalle del Plan
Actor Principal	Usuario Autenticado
Descripción	Permite al usuario ver información detallada de un plan turístico.
Precondiciones	El usuario debe estar autenticado y haber consultado los planes.
Flujo Principal	1. El usuario hace clic en una tarjeta. 2. El sistema carga y muestra la información detallada del plan.
Flujo Alternativo	2a. Si no se encuentra el detalle, mostrar mensaje de error.
Postcondiciones	El usuario visualiza la información detallada.
Requisitos Especiales	Diseño amigable para presentación de datos.

5. Realizar Reserva:

Campo	Descripción
Nombre	Realizar Reserva
Actor Principal	Usuario Autenticado
Descripción	Permite al usuario reservar un plan desde la vista detallada.
Precondiciones	El usuario debe estar autenticado y haber accedido al detalle del plan.

Flujo Principal	1. El usuario hace clic en "Reservar". 2. El sistema registra la reserva y la asocia al usuario.
Flujo Alternativo	2a. Si ocurre un error, mostrar mensaje de error.
Postcondiciones	La reserva queda registrada en la cuenta del usuario.
Requisitos Especiales	Confirmación visual y almacenamiento seguro de la reserva.

6. Ver mis Reservas:

Campo	Descripción
Nombre	Ver Mis Reservas
Actor Principal	Usuario Autenticado
Descripción	Permite al usuario ver un listado de sus reservas previas.
Precondiciones	El usuario debe haber iniciado sesión y tener reservas registradas.
Flujo Principal	1. El usuario accede a "Mis Reservas". 2. El sistema recupera y muestra las reservas.
Flujo Alternativo	2a. Si no hay reservas, mostrar mensaje informativo.
Postcondiciones	El usuario visualiza sus reservas realizadas.
Requisitos Especiales	Interfaz clara para facilitar la consulta.

HISTORIAS DE USUARIO

Aquí se presentan algunas historias de usuario clave para guiar el desarrollo:

1. Registro de Usuario:

- **Como** un visitante,
- **Quiero** poder registrarme en la aplicación usando mi correo electrónico y una contraseña,
- **Para** poder acceder a las funcionalidades de reserva.

2. Inicio de Sesión:

- **Como** un usuario registrado,
- **Quiero** poder iniciar sesión con mi correo electrónico y contraseña,
- **Para** acceder a mi cuenta y realizar reservas.

3. Consulta de Planes:

- **Como** un usuario autenticado,

- **Quiero** ver una lista de planes turísticos disponibles en Colombia (obtenidos de api-Colombia), presentados en tarjetas,
- **Para** poder explorar las opciones disponibles.

4. Ver Detalle del Plan:

- **Como** un usuario autenticado,
- **Quiero** poder hacer clic en una tarjeta de plan turístico,
- **Para** ver información detallada sobre ese plan (descripción, ubicación, etc.).

5. Realizar Reserva:

- **Como** un usuario autenticado,
- **Quiero** poder seleccionar un botón de "Reservar" en la vista detallada de un plan,
- **Para** confirmar mi interés y que la reserva quede registrada en mi cuenta.

6. Ver Mis Reservas (Opcional/Mejora):

- **Como** un usuario autenticado,
- **Quiero** poder ver una lista de los planes turísticos que he reservado,
- **Para** llevar un control de mis viajes planificados.

SEGURIDAD

¿Qué es Snyk?

Snyk es una plataforma de seguridad para desarrolladores que permite a cualquier desarrollador de aplicaciones o en la nube proteger todos los aspectos de su aplicación: buscar y corregir vulnerabilidades desde las primeras líneas de código hasta la nube en ejecución.

¿Cómo funciona Snyk?

Snyk hace pruebas en busca de vulnerabilidades en tu propio código, en dependencias de código abierto, en imágenes de contenedores, en configuraciones de IaC y en entornos de nube. Además, ofrece contexto, prioridades y remedios. (snyk, Analizador código, s.f.)

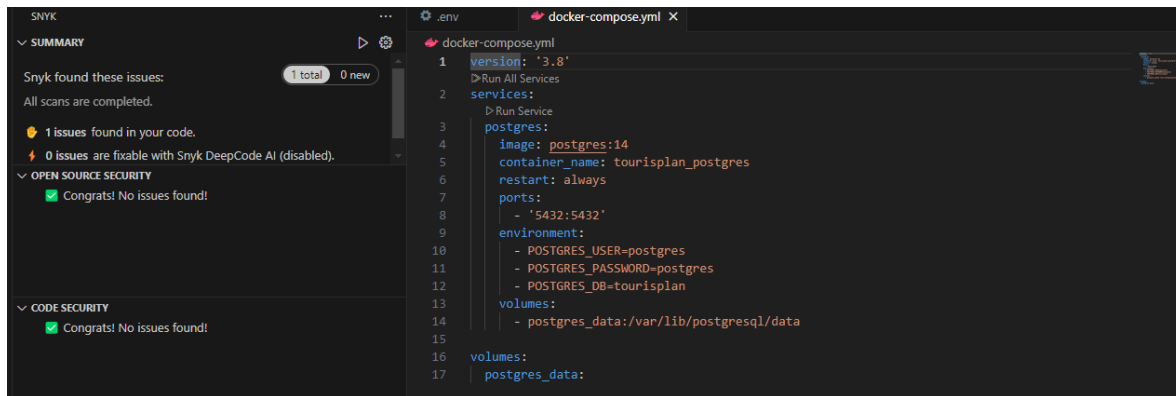


Ilustración 2 escaneo código con Snyk

Como se evidencia en la ilustración número 1 no se evidencia falencia sobre el Código Implementado en el código.

GITHUB

es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para el almacenamiento de repositorios de aplicaciones.

Para nuestro Proyecto escogimos el repositorio GitHub ya que es una plataforma en la nube donde podemos almacenar, compartir y colaborar para escribir código y es la pionera en su función, también nos brinda la facilidad de proporcionar seguridad a nuestro código realizando un escaneo e informando que vulnerabilidades. a continuación, visualizaremos el código en este repositorio.


 ciscojuan	Fusionar solicitud de extracción n.º 65 de ciscojuan/readme	2a2bc57 · hace 1 hora	🕒 108 confirmaciones
flujos de trabajo .github/	Fusionar solicitud de extracción n.º 58 de ciscojuan/feature/...	Hace 2 días	
prisma	Obtenga un plan turístico con identificación obtenida en UU...	ayer	
público	Agregar la siguiente estructura al proyecto	Hace 2 semanas	
origen	Obtenga un plan turístico con identificación obtenida en UU...	ayer	
.dockerignore	Imagen de Docker creada con las configuraciones necesarias...	Hace 7 horas	
.eslintrc	Añadir archivo eslintrc y una clave para el correcto mapeo de ...	Hace 2 semanas	
.gitignore	Inicio de sesión de usuario con credenciales personalizadas, ...	la semana pasada	
Archivo Docker	Imagen de Docker creada con las configuraciones necesarias...	Hace 7 horas	
LICENCIA	Agregar la siguiente estructura al proyecto	Hace 2 semanas	
README.md	Fusionar solicitud de extracción n.º 65 de ciscojuan/readme	hace 1 hora	
docker-compose.yml	docker-compose actualizado	hace 6 horas	
next.config.js	Imagen de Docker creada con las configuraciones necesarias...	Hace 7 horas	
siguiente.config.ts	Imagen de Docker creada con las configuraciones necesarias...	Hace 7 horas	
nextauth.d.ts	Inicio de sesión de usuario con credenciales personalizadas, ...	la semana pasada	
paquete-lock.json	Se aplicaron cambios de AmazonQ	Hace 2 días	

Ilustración 3 Código GitHub

DEPENDABOT

☐

🕒

2 Abierto

✓

0 Cerrado

Paquete ▾

Ecosistema ▾

Manifiesto ▾

Gravedad ▾

Clasificar ▾

☐

🕒

libxmljs2 es vulnerable a la confusión de tipos al analizar XML especialmente diseñado

#1 abierto 2 days ago • Detectado en libxmljs2 (npm) • package-lock.json

Crítico

Desarrollo

☐

🕒

La cookie acepta el nombre de la cookie, la ruta y el dominio sin caracteres fuera de los límites

#2 abierto last week • Detectado en cookie (npm) • package-lock.json

Bajo

🔍

Consejo: Usa `hasvulnerable-calls` para ver alertas con llamadas a funciones vulnerables

🔗 Consejo: Usa [has:vulnerable-calls](#) para ver alertas con llamadas a funciones vulnerables.

Ilustración 4 Análisis Vulnerabilidad GitHub

Como podemos observar en la Ilustración 4 observamos dos vulnerabilidades una en estado Crítico y en estado Bajo.

VULNERABILIDAD CRITICA

libxmljs2 es vulnerable a una vulnerabilidad de confusión de tipos al analizar un XML especialmente diseñado al invocar la namespaces()función (que invoca XmlNode::get_local_namespaces()) en un nodo secundario de un nodo que hace referencia a una entidad. Esta vulnerabilidad puede provocar denegación de servicio y ejecución remota de código.

GRAVEDAD

Crítico

9.2

/ 10

Métricas base de CVSS v4

Métricas de explotabilidad

Vector de ataque Red

Complejidad del ataque Bajo

Requisitos de ataque Presente

Privilegios requeridos Ninguno

Interacción del usuario Ninguno

Métricas de impacto del sistema vulnerable

Confidencialidad Alto

Integridad Alto

Disponibilidad Alto

Métricas de impacto del sistema subsiguientes

Confidencialidad Ninguno

Integridad Ninguno

Disponibilidad Ninguno

Puntuación EPSS

(percentil 54)

Debilidades

CWE-843

Identificación CVE

CVE-2024-34394

Identificación de GHSA

GHSA-78h3-pg4x-j8cv

VULNERABILIDAD EN SYLIUS (CVE-2024-34349)

Sylius es una plataforma de comercio electrónico de código abierto. Antes de 1.12.16 y 1.13.1, existe la posibilidad de ejecutar código JavaScript en el panel de administración. Para realizar un ataque XSS, ingrese un script en el campo Nombre en cuál de los recursos: Taxones, Productos, Opciones de producto o Variantes de producto. El código se ejecutará mientras se utiliza un campo de autocompletar con una de las entidades enumeradas en el Panel de administración. También para los taxones en el árbol de categorías en el formulario del producto. El problema se solucionó en las versiones: 1.12.16, 1.13.1.

VULNERABILIDAD BAJA

La cookie acepta el nombre de la cookie, la ruta y el dominio sin caracteres fuera de los límites #2

IMPACTO

El nombre de la cookie podría usarse para configurar otros campos de la cookie, lo que generaría un valor de cookie inesperado. Por ejemplo, `serialize("userName=<script>alert('XSS3')</script>; Max-Age=2592000; a", value)` esto resultaría en `"userName=<script>alert('XSS3')</script>; Max-Age=2592000; a=test"`, configurando `userName` la cookie como `<script>` e ignorando `value`.

Se puede utilizar un escape similar para `pathy domain`, que podría utilizarse de forma abusiva para alterar otros campos de la cookie.

PARCHES

Actualice a 0.7.0, que actualiza la validación para `name`, `path`, y `domain`.

SOLUCIONES ALTERNATIVAS

Evite pasar valores no confiables o arbitrarios para estos campos, asegúrese de que sean configurados por la aplicación en lugar de la entrada del usuario.

Debilidades

DebilidadCWE-74

Identificación CVE

CVE-2024-47764

Identificación de GHSA

GHSA-pxg6-pf52-xh8x

GitHub Code Secutity

Una de las características que ofrece GitHub para escanear el código del proyecto en tiempo real, automatizando la integración y despliegue continuo CI /CD.

Al igual que en el caso anterior, code security al detectar una vulnerabilidad de código, nos muestra el script que disparó esa alerta y nos sugiere una remediación.

Speed up the remediation of this alert with [Copilot Autofix for CodeQL](#)

Generate fix

.github/workflows/ci.yml:56

```
53     run: npm run build
54
55     type-check:
56       name: TypeScript Check
57       runs-on: ubuntu-latest
58       steps:
59         - name: Checkout code
60           uses: actions/checkout@v4
61
62         - name: Setup Node.js
63           uses: actions/setup-node@v4
64           with:
65             node-version: "20"
66             cache: "npm"
67
68         - name: Install dependencies
69           run: npm ci
70
71         - name: Check TypeScript
72           run: npx tsc --noEmit
```

Actions job or workflow does not limit the permissions of the GITHUB_TOKEN. Consider setting an explicit permissions block, using the following as a minimal starting point: {contents: read}

CodeQL

Archivo ci.yml workflow de GitHub Actions

.github/workflows/ci.yml:56-72Autofix

Warning

Workflow does not contain permissions

Actions job or workflow does not limit the permissions of the GITHUB_TOKEN. Consider setting an explicit permissions block, using the following as a minimal starting point: {contents: read}

If a GitHub Actions job or workflow has no explicit permissions set, then the repository permissions are used. Repositories created under organizations inherit the organization permissions. The organizations or repositories created before February 2023 have the default permissions set to read-write. Often these permissions do not adhere to the principle of least privilege and can be reduced to read-only, leaving the write permission only to a specific types as issues: write or pull-requests: write.

... 00 -15,2 +15,5 00

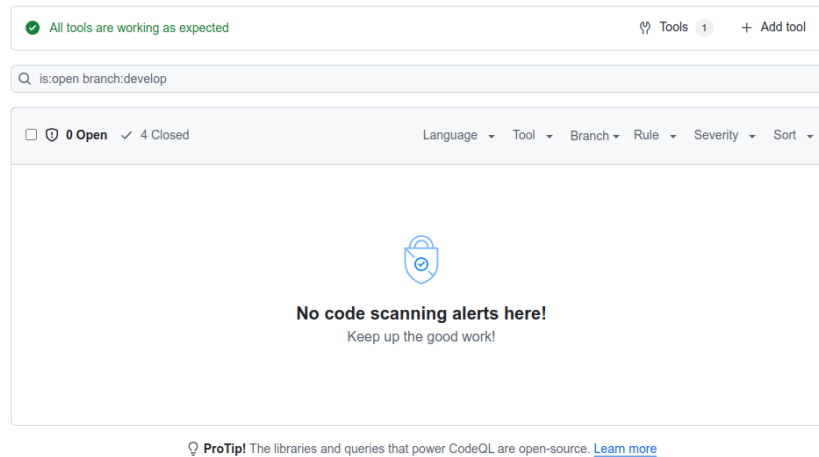
15	15
	16 permissions:
	17 contents: read
	18
16 jobs:	19 jobs:

Copilot Autofix for CodeQL is powered by AI and may make mistakes. Always verify output.

Commit to new branch

Remediación sugerida por GitHub

Code scanning

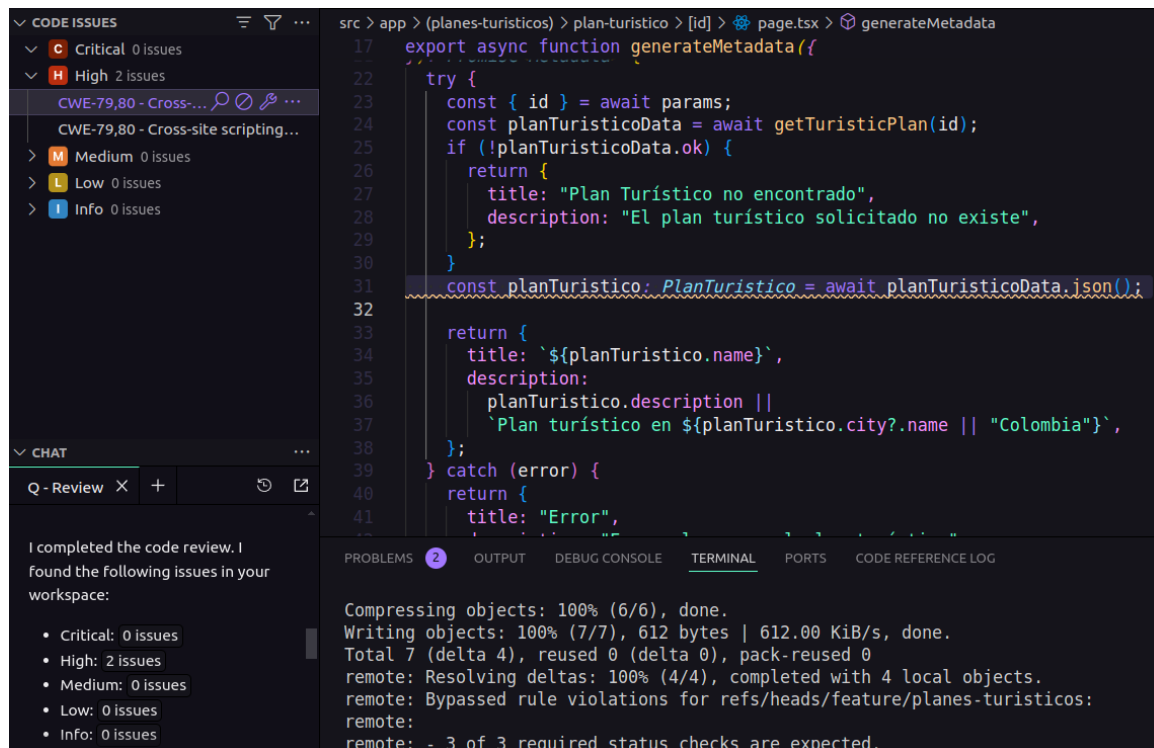


Análisis posterior luego de implementar las remediaciones sugeridas.

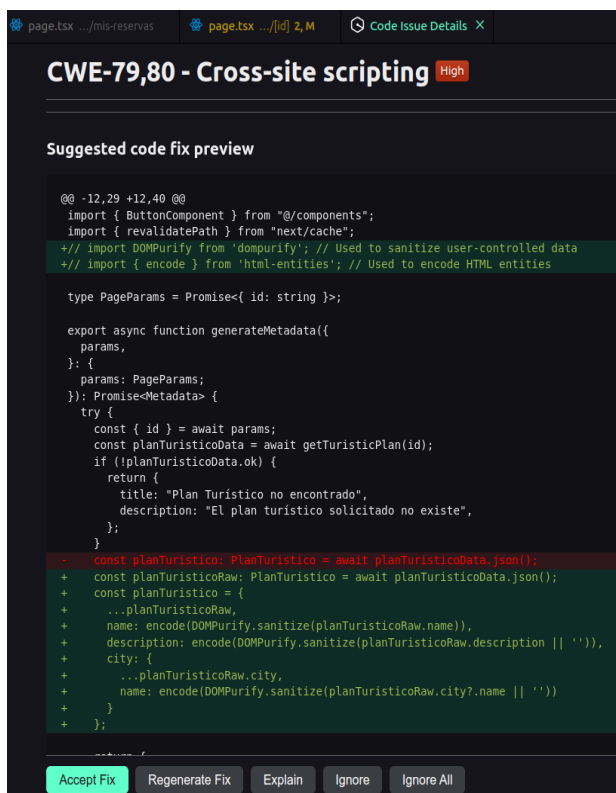
Amazon Code Review

Es una herramienta que nos permite revisar el código del proyecto para en busca de alguna vulnerabilidad en tiempo real. Luego del análisis, si detecta alguna vulnerabilidad, Amazon Q nos da recomendaciones para mitigar dicha vulnerabilidad.

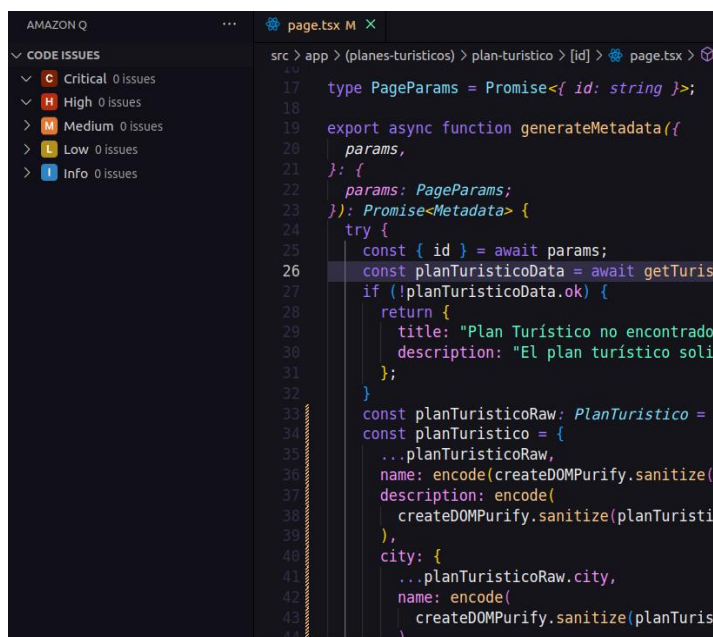
Usando Amazon Q



El resultado del análisis permitió descubrir 2 vulnerabilidades de tipo XSS con una gravedad alta como se puede observar en la imagen.



Amazon Q nos da una recomendación para remediar la vulnerabilidad encontrada



Este fue el análisis posterior luego de realizar los cambios sugeridos por Amazon.