

TRABAJO DE CALIDAD DE SOFTWARE

Informe Final: Droguería Sar

Nombre del aprendiz:

Camilo Andrey Diaz Sarmiento

Nombre del instructor:

Javier Leonardo Pineda

Institución:

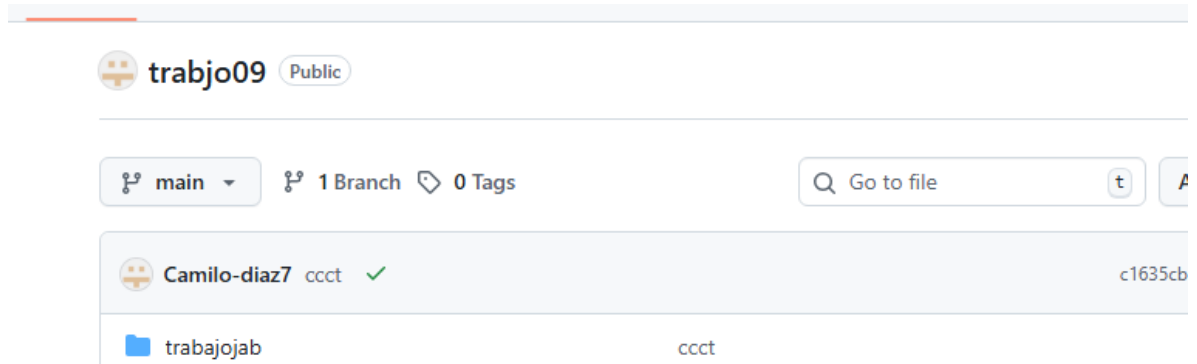
SENA centro de servicios financieros

Fecha de creación:

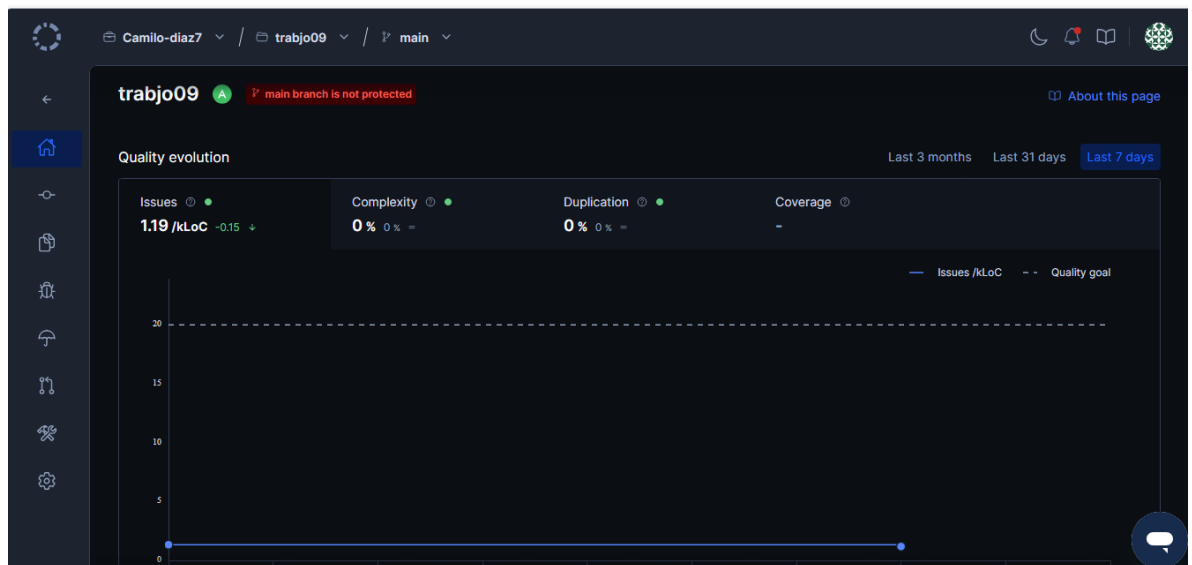
19/10/2025

TRABAJO DE CALIDAD DE SOFTWARE

1. Creación del repositorio en GitHub



1.1. Autenticación del repositorio en codacy para su respectivo escaneo



1.1. Seguridad

Errores de seguridad

TRABAJO DE CALIDAD DE SOFTWARE

Issues breakdown

27 total issues

Category		Total
Security	<div><div></div></div>	18
Error prone	<div><div></div></div>	9

[See all issues](#)

 MEDIUM

Error prone

Operator ! prohibited; use === FALSE instead

trabajojob/trabajo/routes/web.php

```
19 // Función de verificación de admin
20 $checkAdmin = function () {
21     if (!Auth::check() || !Auth::user()->is_admin) {
22         return redirect('/')->with('error', 'Acceso no autorizado');
23     }
24 }
```



Introduced by Camilo-díaz7, 2 days ago



Time to fix: 5 minutes



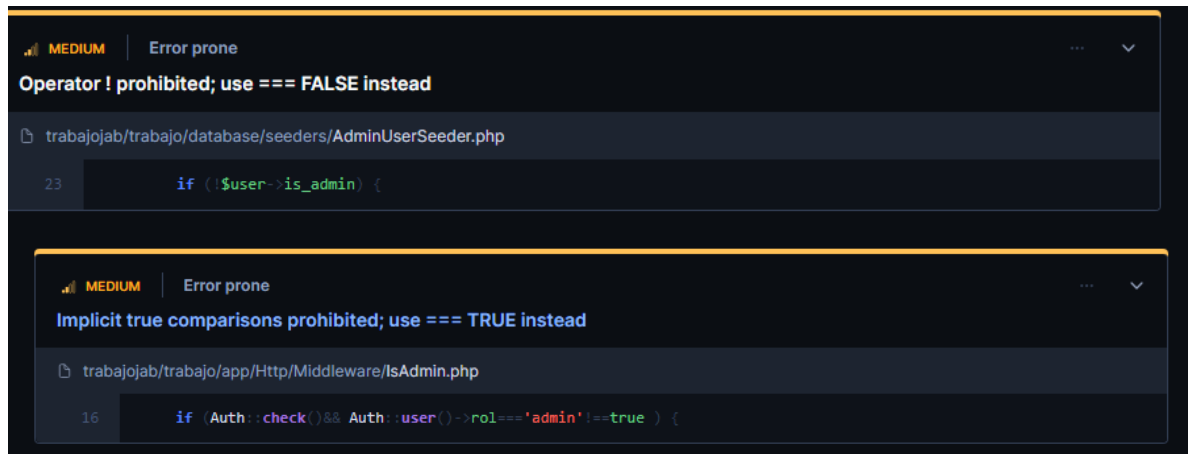
[Create Jira ticket](#)

[About this pattern](#) [PHP_CodeSniffer's documentation](#)

Why this is a problem

Incorrect or inconsistent use of comparison operators can lead to unexpected behavior or bugs, especially when using loose comparison operators that may cause type juggling issues.

TRABAJO DE CALIDAD DE SOFTWARE



2. Solución de errores

2.1. Se implemento la recomendación de codacy del error leve y ya no nos aparece al hacer el escaneo y se determinó que es una mala practica que afecta en el sistema lo cual impacta al saber si al hacer el chequeo del controlador del usuario no tuviera un dato booleano nos daba resultados aleatorios cuando entra el usuario puede entrar si es cliente o más por eso se utiliza el método falso y verdadero

```
...  
...  
$checkAdmin = function (): RedirectResponse {  
    if (Auth::check() !== false || Auth::user()->is_admin !== true) {  
        return redirect(to: '/')->with(key: 'error', value: 'Acceso no autorizado');  
    }  
};
```

2.2. La mala practica en este código fue no ver puesto que si el dato no es verdadero nos

TRABAJO DE CALIDAD DE SOFTWARE

guardaba en el sistema o computadora por
es se implemento ese ti de dato booleano

```
if (!$user->is_admin !== true) {  
    $user->save();  
}
```

2.3. En esta parte del codigo el usuario lo determinamos en objeto para cuando el dato que no lanza el admin sea verdadero nos mande a su respectiva pagina

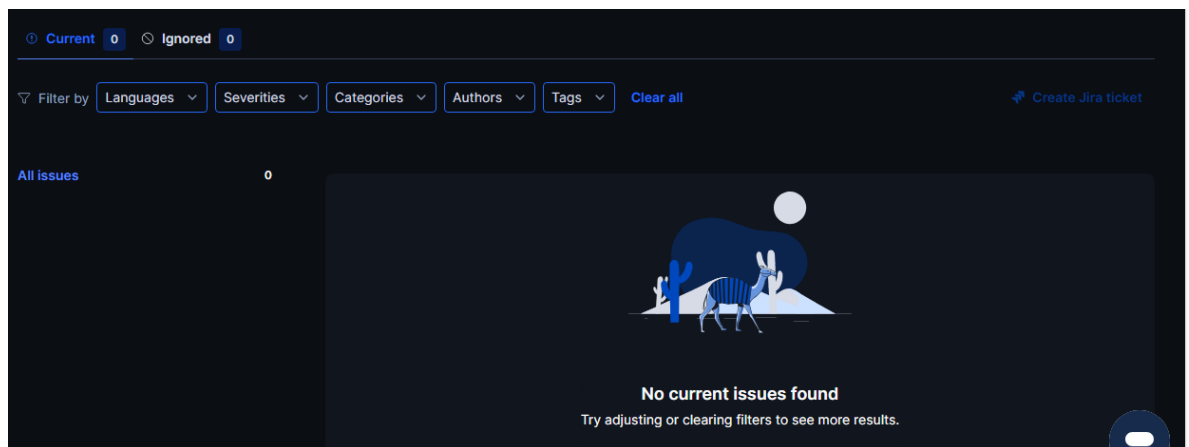
```
{  
    $user = Auth::user();  
  
    if (is_object(value: $user) && $user->is_admin === true) {  
        return route(name: 'admin.dashboard');  
    }  
    return '/home';  
}
```

2.4.

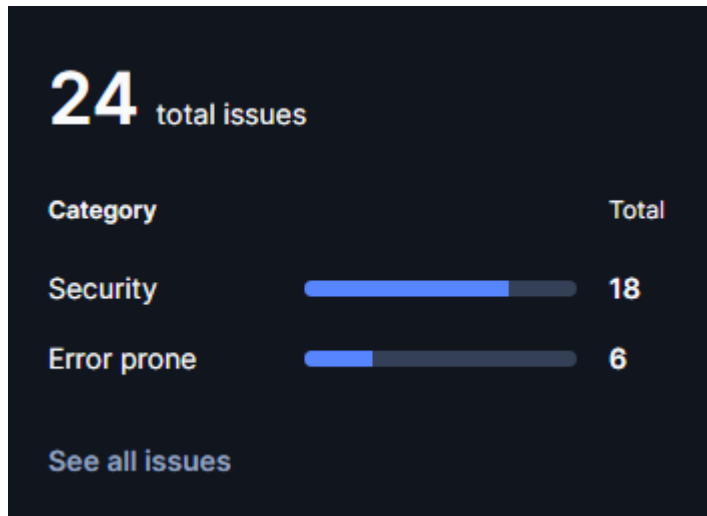
Error detectado	Descripción	Antes	Después	Solución aplicada
La regla prohíbe ! por motivos de legibilidad y seguridad: recomienda comparaciones estrictas	Se usan malas practicas en el código lo que nos da ambigüedades y coerción de tipos	<pre>if (!Auth::check() !Auth::user()->is_admin) { return redirect('/')-</pre>	<pre>\$checkAdmin = function () { if (Auth::check() === false Auth::user()->is_admin !== true) { return redirect('/')-</pre>	Poner ese condicional para dar una buena practica del código

TRABAJO DE CALIDAD DE SOFTWARE

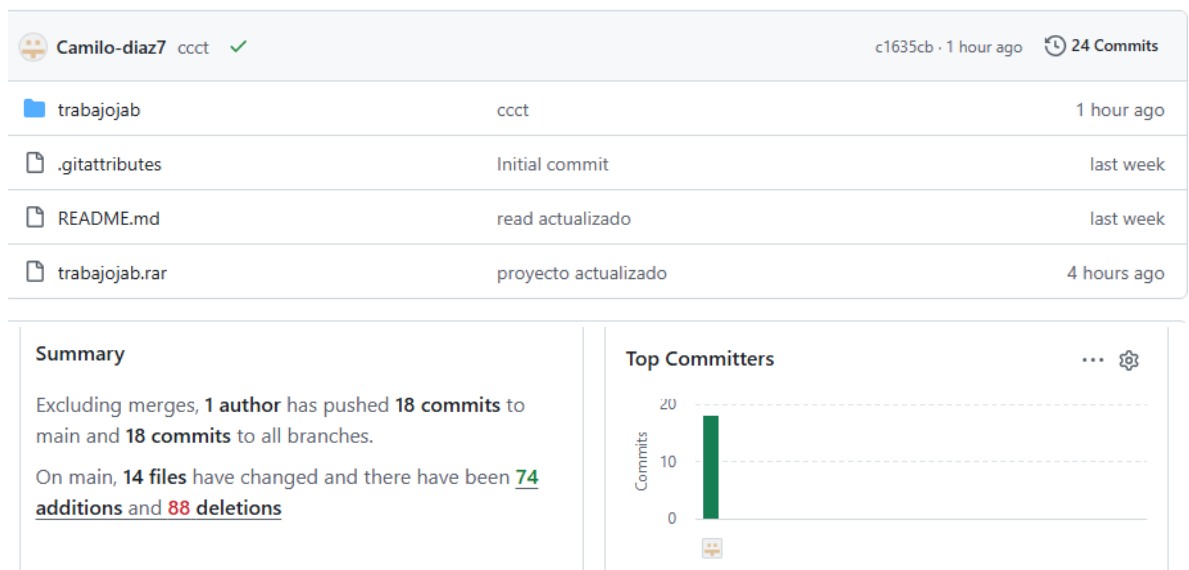
		<pre>>with('error', 'Acceso no autorizado'); } };</pre>	<pre>>with('error', 'Acceso no autorizado'); } };</pre>	
violación de una regla de estilo del analizador estático en este caso en los seeders	Se usan malas practicas en el código lo que nos da ambigüedades y coerción de tipos	<pre>if (!\$user->is_admin) { \$user->true \$user->save(); }</pre>	<pre>if (!\$user->is_admin !== true) { \$user->save(); }</pre>	La solución aplicada fue implementar el negador lógico de posesión y unirlo con el tipo de dato booleano
comparación implícita con true o una expresión confusa de validación booleana	Nos indica una implícita lo que nos deja, lo cual genera ambigüedad para el analizador.	<pre>if (Auth::check() && Auth::user()->rol === 'admin' !== true) {</pre>	<pre>if (Auth::check()===false && Auth::user()->rol==='admin') {</pre>	La solución aplicada fue autenticar al usuario y si no es admin bloquearlo



TRABAJO DE CALIDAD DE SOFTWARE



3. Comits y archivos en el repositorio trabajo09 del tema de la droguería



4. Reflexión personal: Los errores de la mala práctica de codacy me ayudaron en analizar el código más a fondo para que no haya bugs y vulnerabilidades que puedan afectar al proyecto

TRABAJO DE CALIDAD DE SOFTWARE