*"Año del Bicentenario del Perú: 200 años de Independencia"*

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

*Escuela profesional de Ingeniería de Software*

**

**GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE - G1**

***1er Testeo del proyecto MM&P: Tienda virtual***

**INTEGRANTES:**

Chavez Burgos, Luiz Arnold (19200250)

López Loaiza, Edgar Fernando (19200295)

Huamán Ampuero, Lucero Marysol (19200081)

Magallanes Quiroz, Claudia Carolina (18200331)

Quispe Alarcon, Jorge Luis (19200094)

Romero Angeles, Luis Alfredo Felix (19200317)

Zafra Moran, Rolando Jesus (19200262)

Zarate Villar, Jhennyfer Nayeli (19200248)

Calderon Herrera Miguel Angel (19200071)

Tomasto Solis, Victor Eduardo (18200299)

**DOCENTE RESPONSABLE:**

Prof. Espinoza Robles, Armando David

ÍNDICE

[**Dashboard utilizado**](#_nffum4dpjxxl) **3**

[**Monitor de seguridad**](#_j7e88uo68wei) **4**

[Auth](#_vszi9pdywwpt) 4

[Command Injection](#_7xl9kr1mboqz) 5

[Cryptography](#_x0vnlkkbzvn9) 5

[Insecure Modules Libraries (Biblioteca de módulos inseguros)](#_53orhbp7wea) 6

[Insecure Storage](#_v4qbg4dvglvd) 6

[Malicious Code (Código malicioso)](#_ljgoddtljiuq) 6

[Regex](#_94jrhi823xhq) 7

[Routes (Rutas)](#_80fgnm71dtm5) 7

[SQL Injection](#_izt46knb6o6h) 7

[Unexpected Behaviour (Comportamiento inesperado):](#_w9oko49p7dhw) 7

[Visibility (Visibilidad)](#_oe3k34h73usx) 8

[XSS](#_o6x8qgk6hh72) 8

[**Code Patterns**](#_3dsjo4tuek3j) **8**

[**Issues**](#_v616tz6qb945) **8**

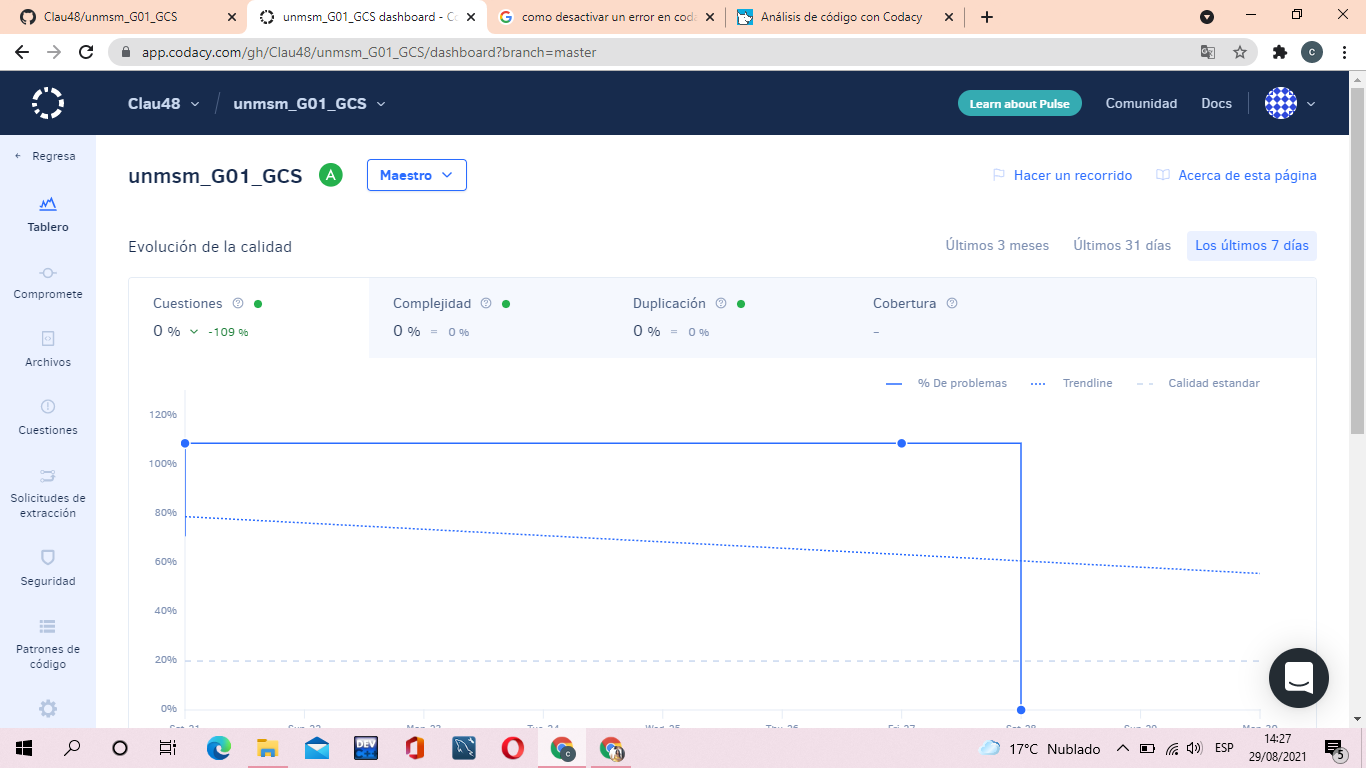
[**Archivos**](#_j4e6x2odfpet) **8**

Desarrollo del Documento

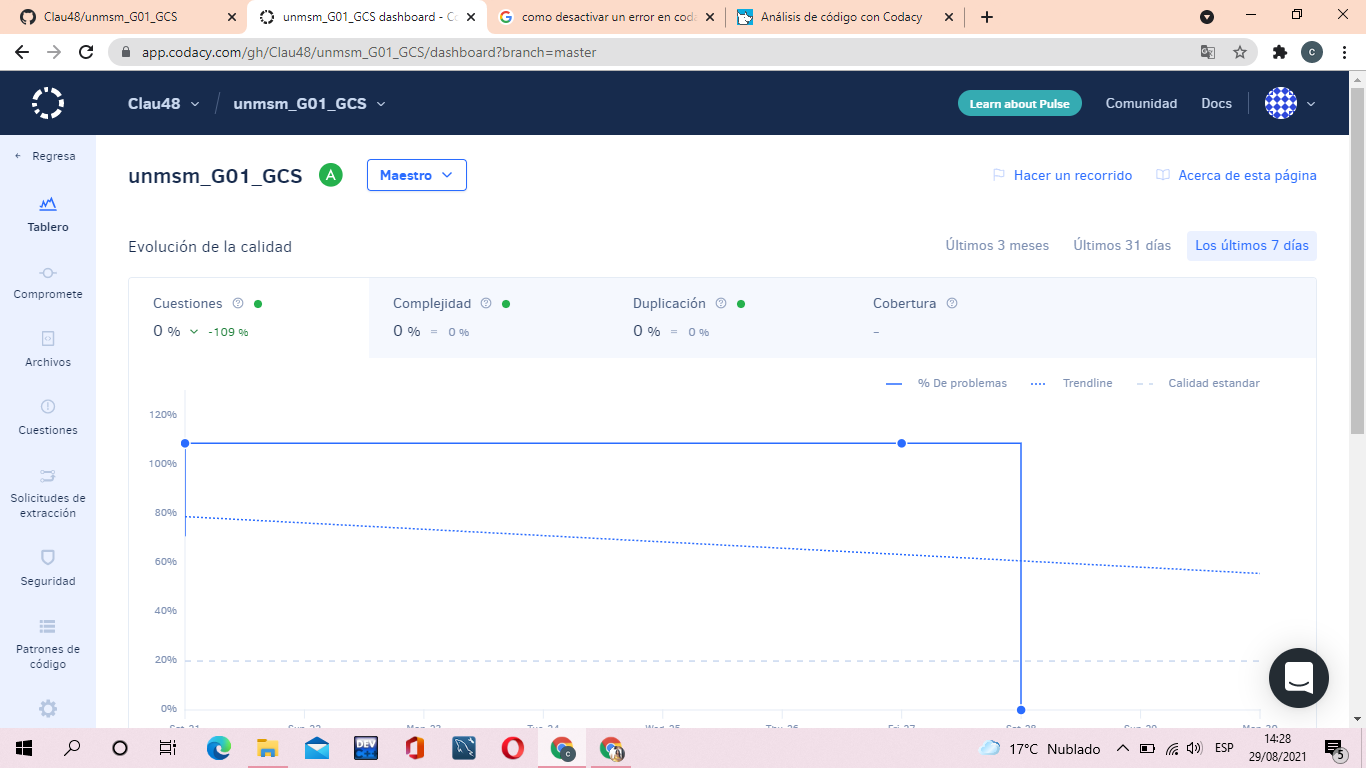
# Dashboard utilizado

Codacy es una herramienta que nos ayuda al control de la deuda técnica y a mejorar la calidad y producción de nuestros equipos de desarrollo.

En el análisis actual nos brindó una calificación general del código: A. Anteriormente nos daba una calificación de C, ya que dentro del código se encontraban los módulos usados en la base de datos, pero al retirarlos la calificación cambió al actual.



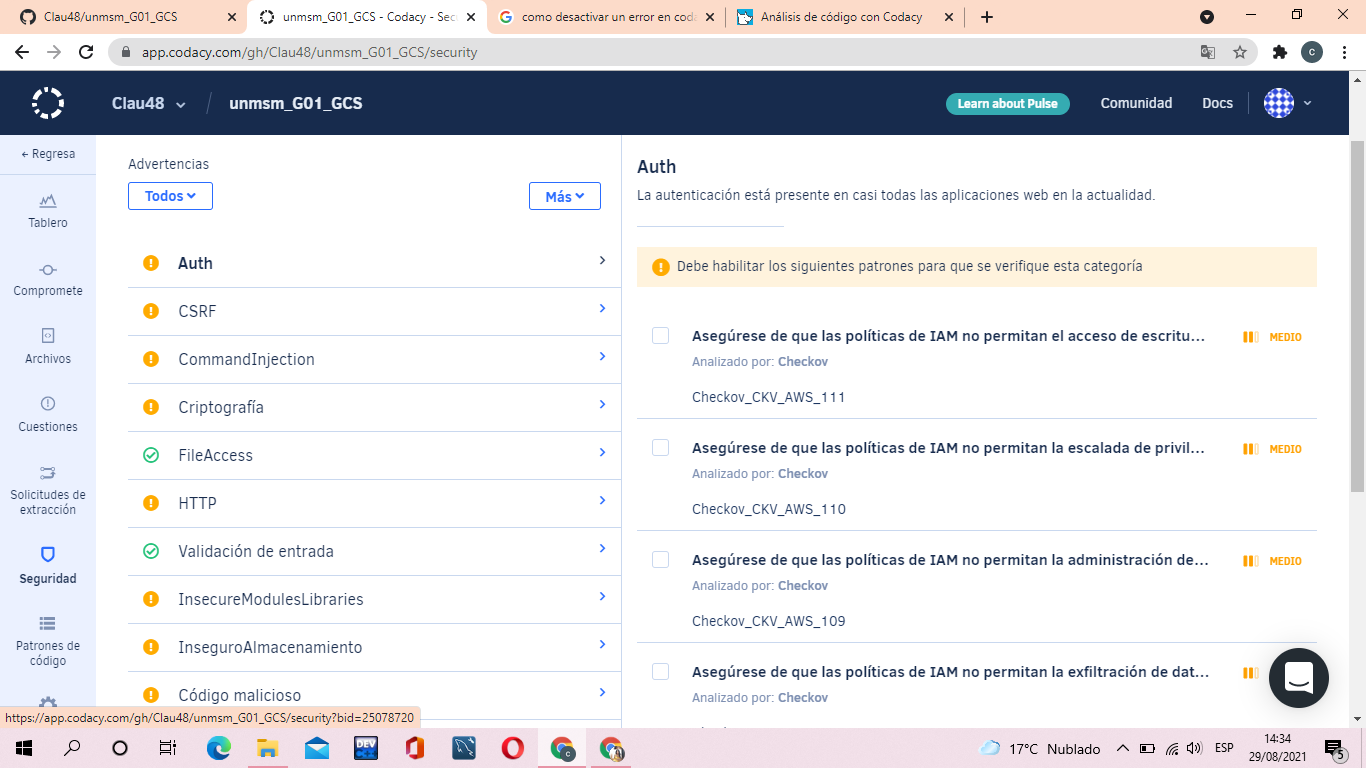
Tenemos una evolución de calidad con un porcentaje de cuestionamientos de 0%, complejidad de 0% y duplicación de 0%.



Aquí se puede visualizar de manera gráfica el avance del código a través de pull en la rama master, mostrando los problemas y la calidad estándar del código.



# Monitor de seguridad





## Auth

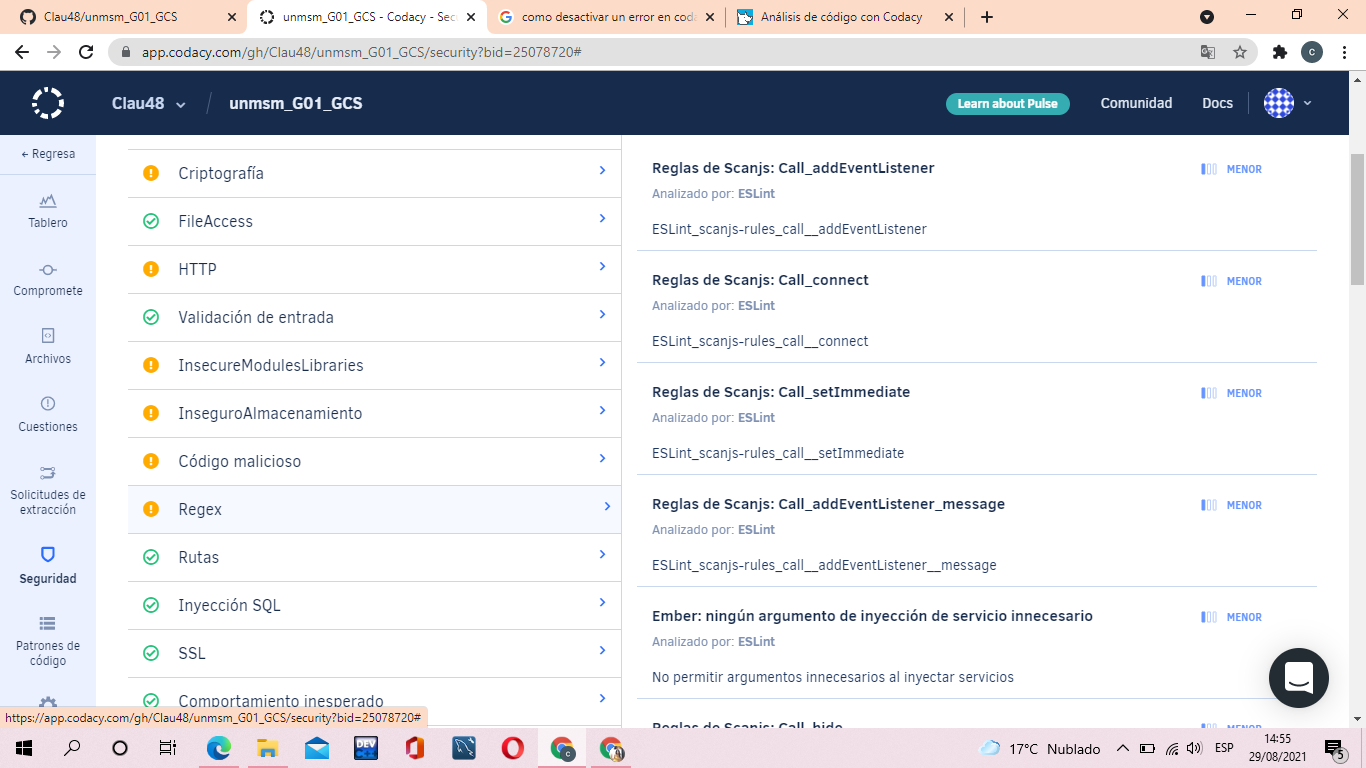
Autenticación que está presente en casi todas las aplicaciones web en la actualidad. Los problemas encontrados aquí son:

* 1. Asegurarse de que las políticas de IAM no permitan la administración de permisos.
  2. Asegurarse de que las políticas de IAM no permitan la exposición de credenciales.
  3. Asegurarse de que las políticas de IAM no permitan el acceso de escritura sin restricciones.
  4. Asegurarse de que las políticas de IAM no permitan la escalada de privilegios.
  5. Asegurarse de que las políticas de IAM no permitan la exfiltración de datos

Todos estos problemas tienden a tener acciones de exposición de los recursos del programa calificándolos en nivel intermedio.

## Command Injection

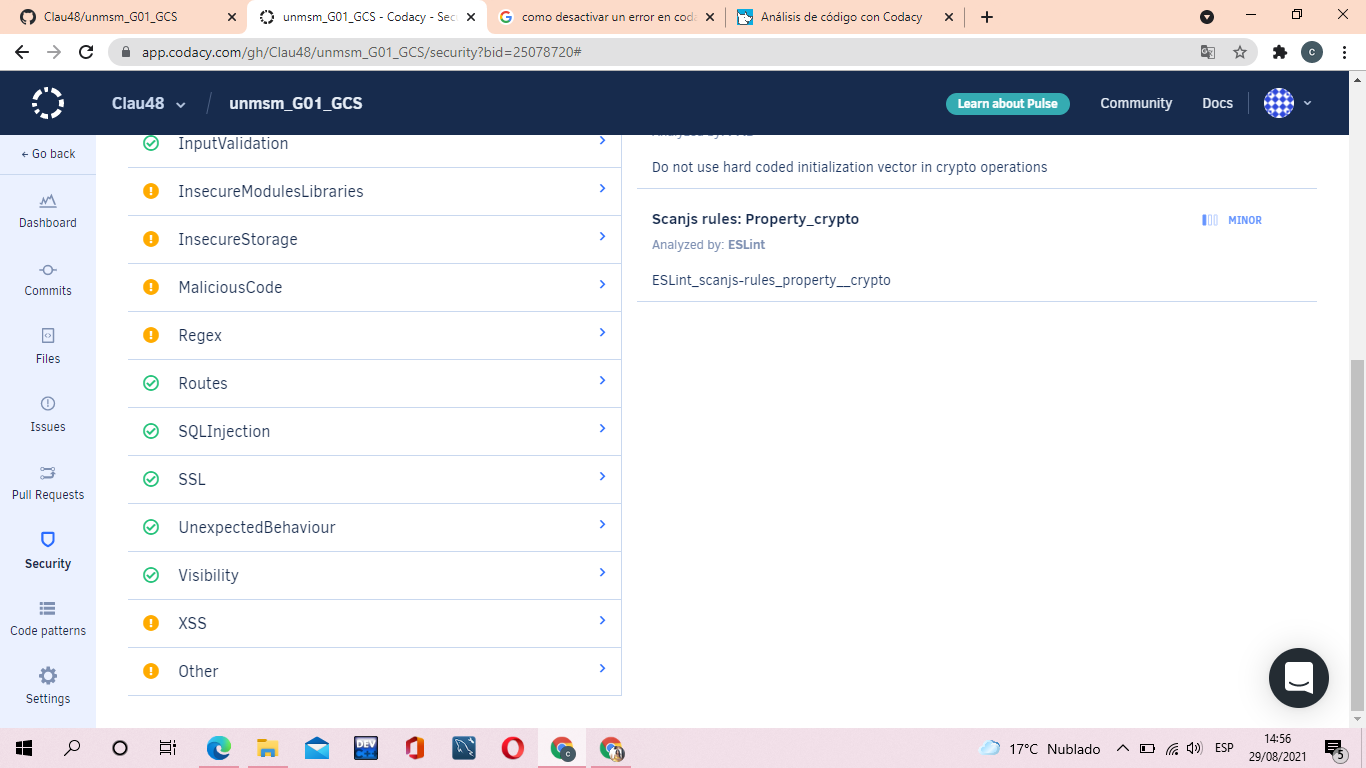
Refiere a la inyección de comandos es un ataque en el que el objetivo es la ejecución de comandos arbitrarios en el sistema operativo host. En este aspecto se encontraron problemas de rango menor en el código del programa.



## Cryptography

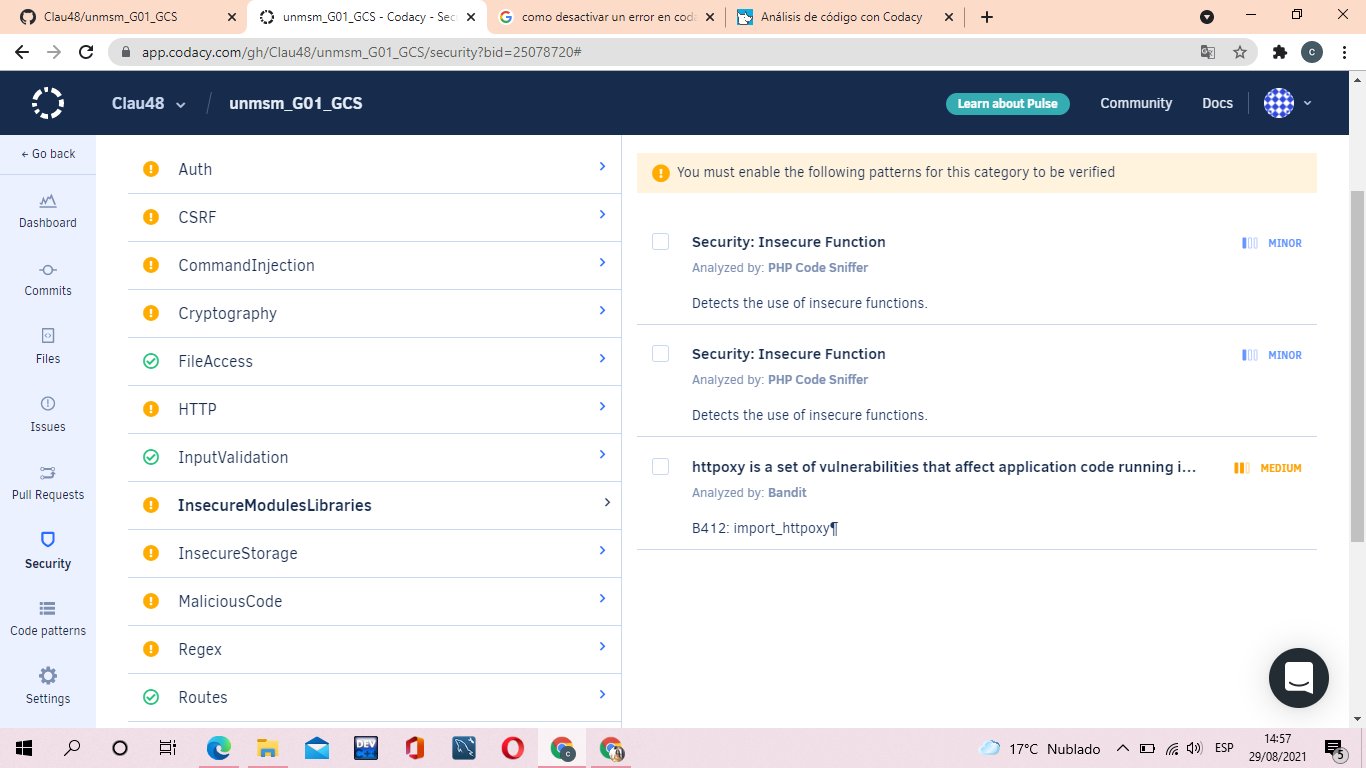
Es una técnica de seguridad muy utilizada y existen varias funciones criptográficas, pero no todas son seguras.Entre sus problemas tenemos:

* 1. Clave criptográfica codificada: Siendo calificado como algo crítico,ante esto no se utilizan claves de cifrado codificadas de forma rígida.
  2. Crypto IV inseguro: Se califica como algo crítico, ante ello no se utiliza un vector de inicialización codificado de forma rígida en operaciones criptográficas.
  3. Reglas de Scanjs(Property\_crypto) : Siendo calificado como algo menor.



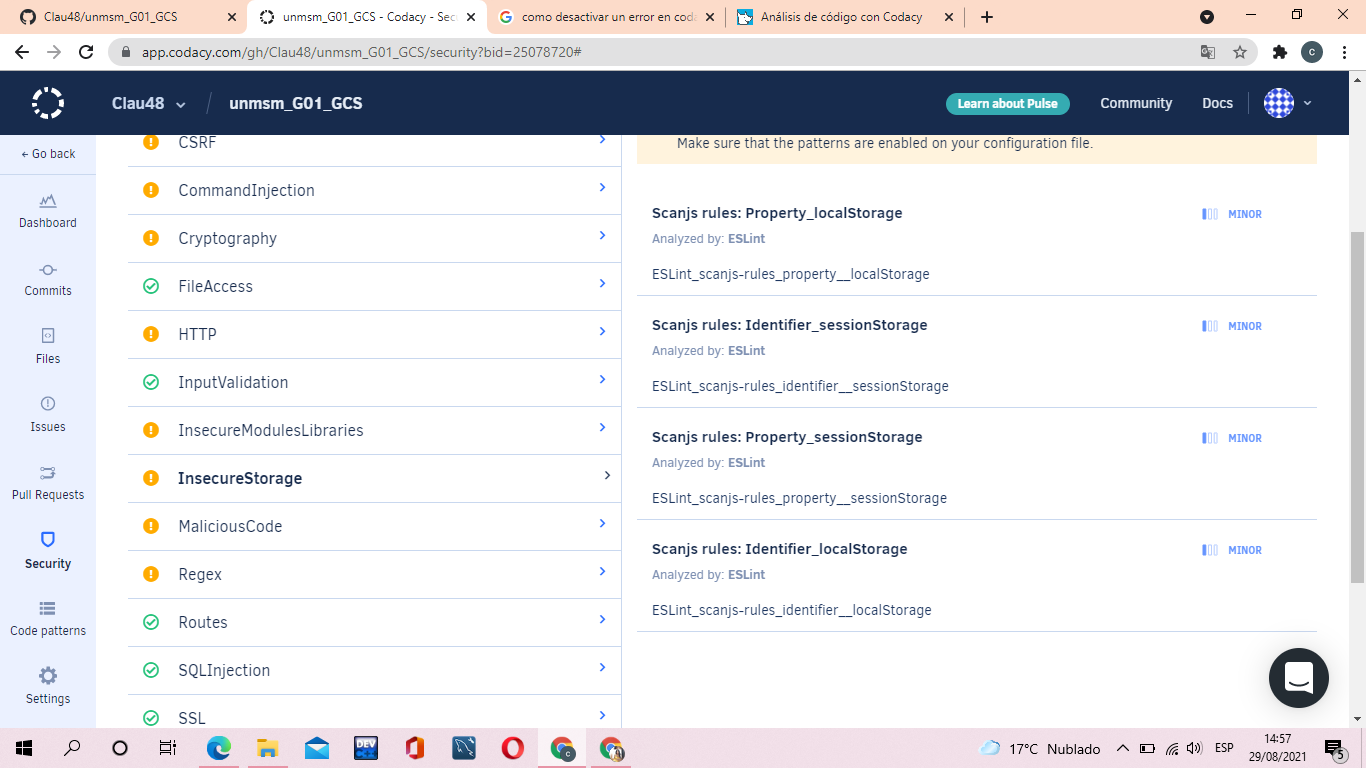
## Insecure Modules Libraries (Biblioteca de módulos inseguros)

Considera las posibles implicaciones de seguridad asociadas con algunos módulos.En este ámbito no se encuentran problemas.



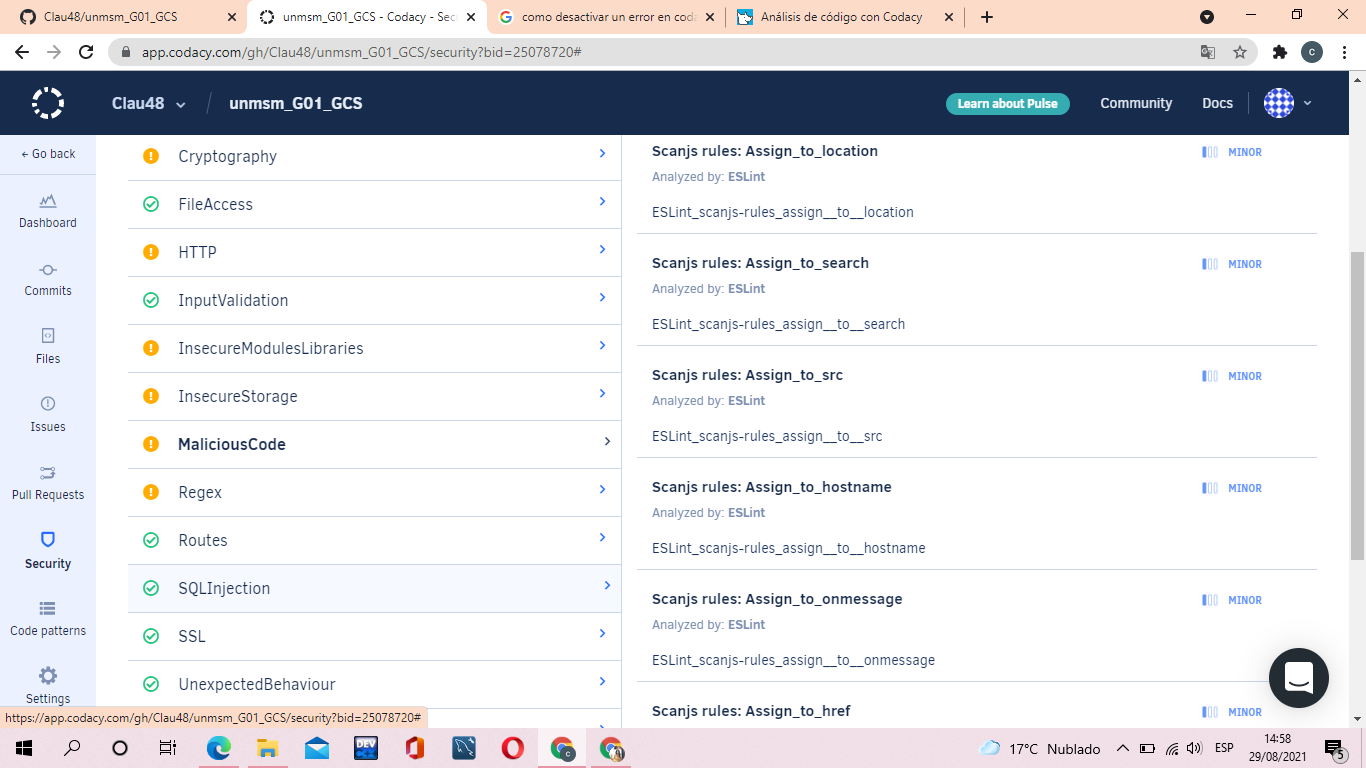
## Insecure Storage

Refiere a que el almacenamiento de datos confidenciales con estas API no es seguro.El problema de calificación menor es el habilitar los patrones para que verifique esta categoría:



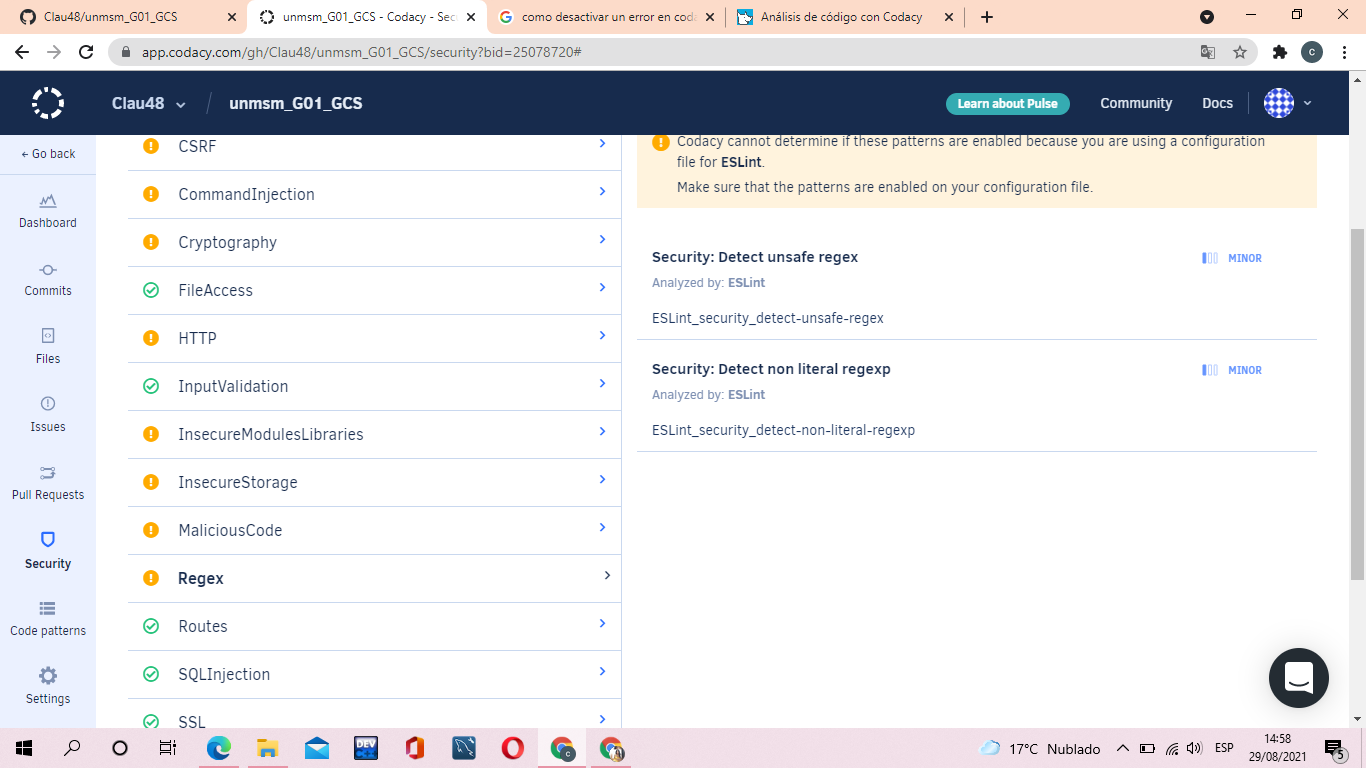
## Malicious Code (Código malicioso)

Este problema se refiere a que se puede acceder a las API internas expuestas o modificarlas mediante un código malicioso o por accidente desde otro paquete.Estos problemas son de calificación menor el cual requiere habilitar algunos patrones para verificar las siguientes categorías:



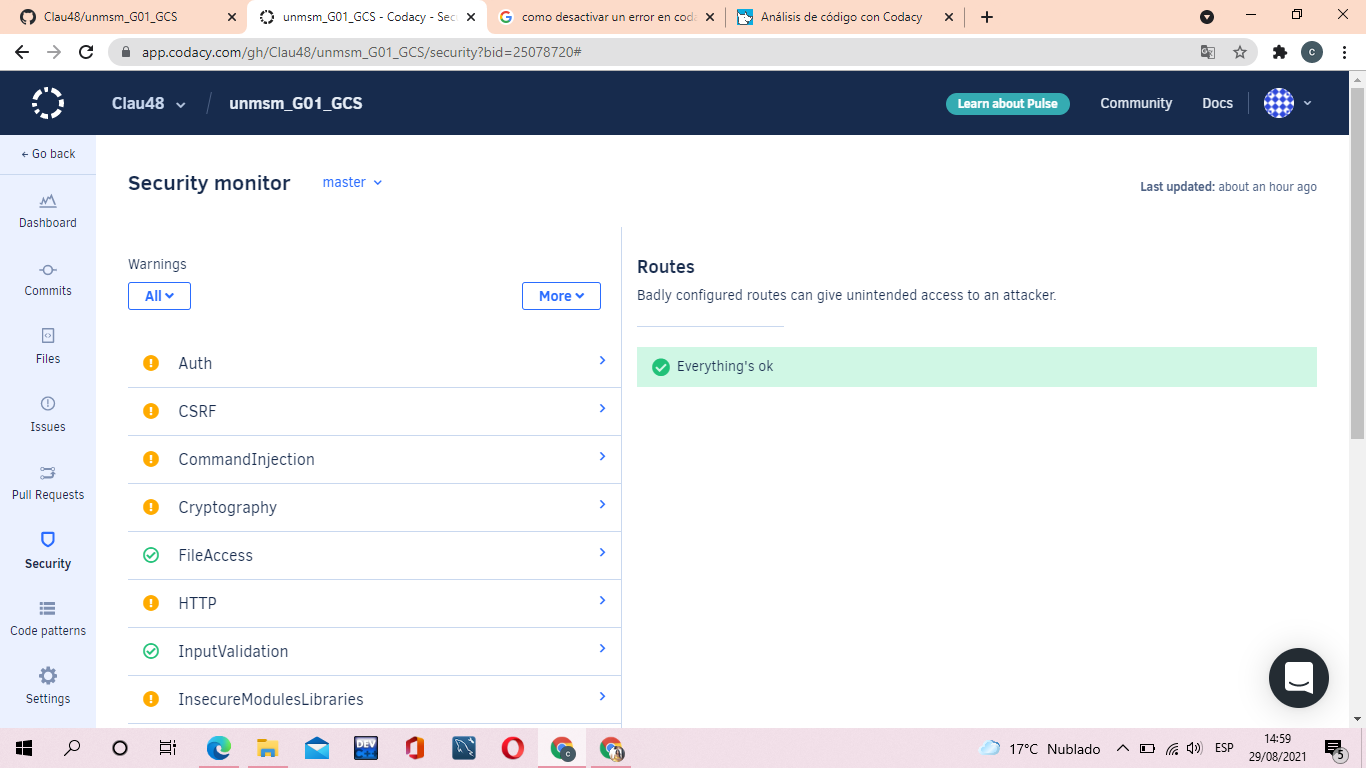
## Regex

Se puede usar en un ataque de denegación de servicio.no existen problemas en este ámbito.



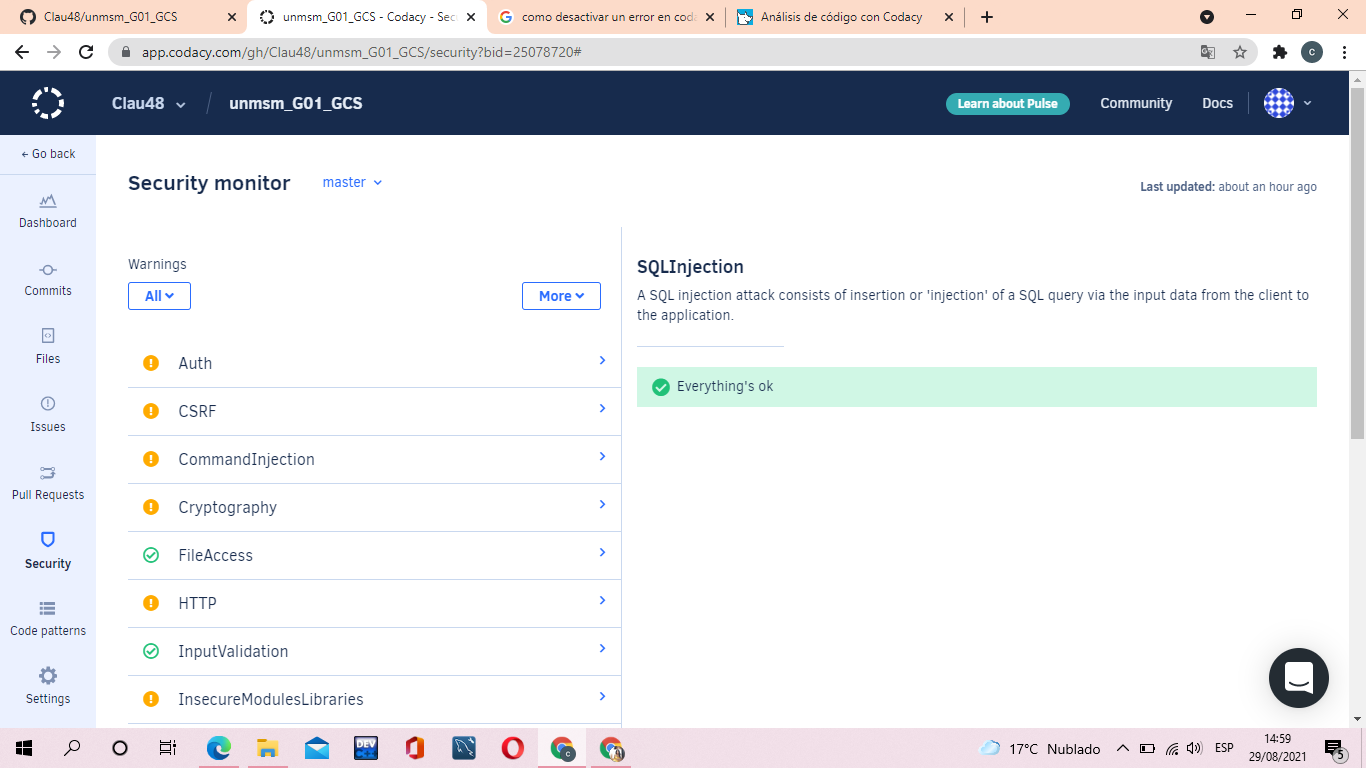
## Routes (Rutas)

Si están mal configuradas pueden dar acceso no deseado a un atacante.No existen problemas en este ámbito.



## SQL Injection

Su ataque consiste en la inserción de una consulta SQL a través de los datos de entrada del cliente a la aplicación. No existen problemas en este ámbito.



## Unexpected Behaviour (Comportamiento inesperado):

En este ámbito todo se encuentra correcto.

## Visibility (Visibilidad)

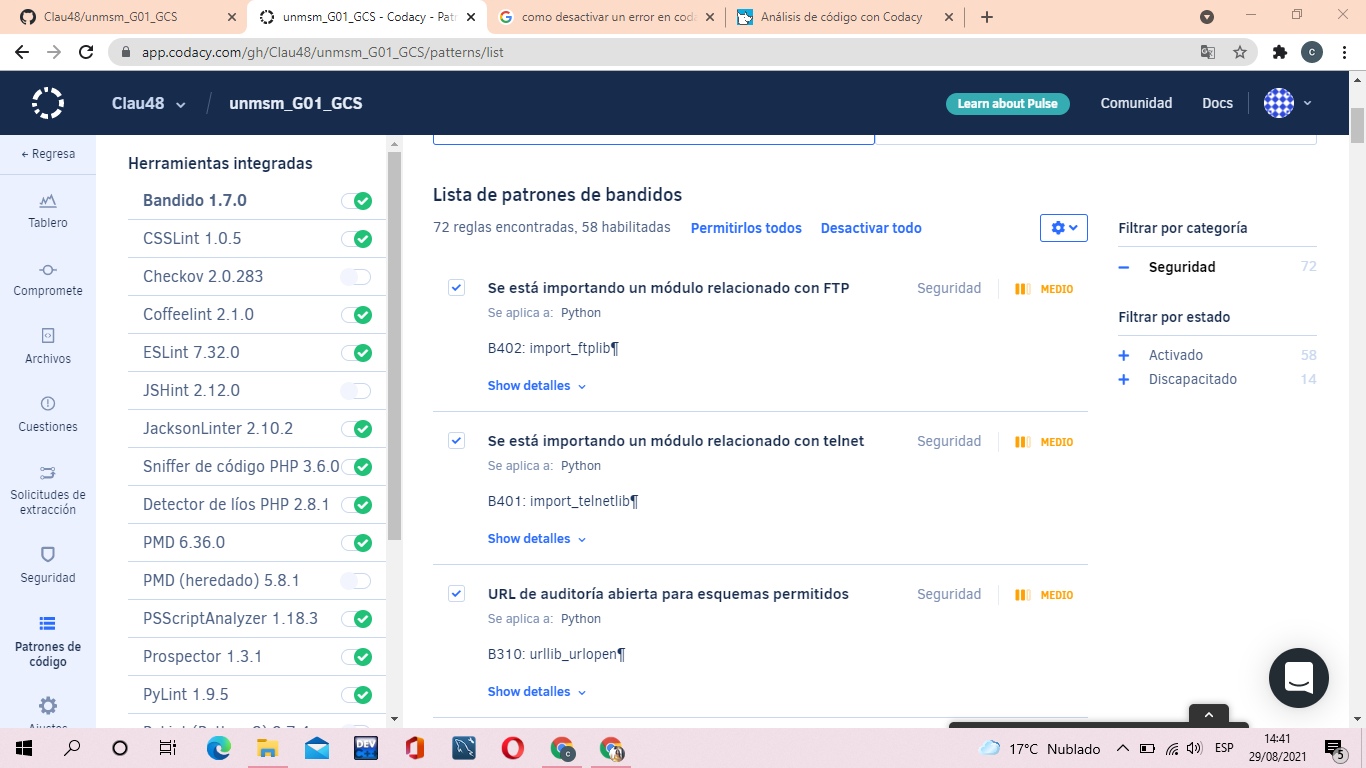
Referido a los campos que no deben tener acceso público. Todo se encuentra correcto en este ámbito

## XSS

Permite a los atacantes inyectar scripts del lado del cliente en las páginas web que ven otros usuarios.

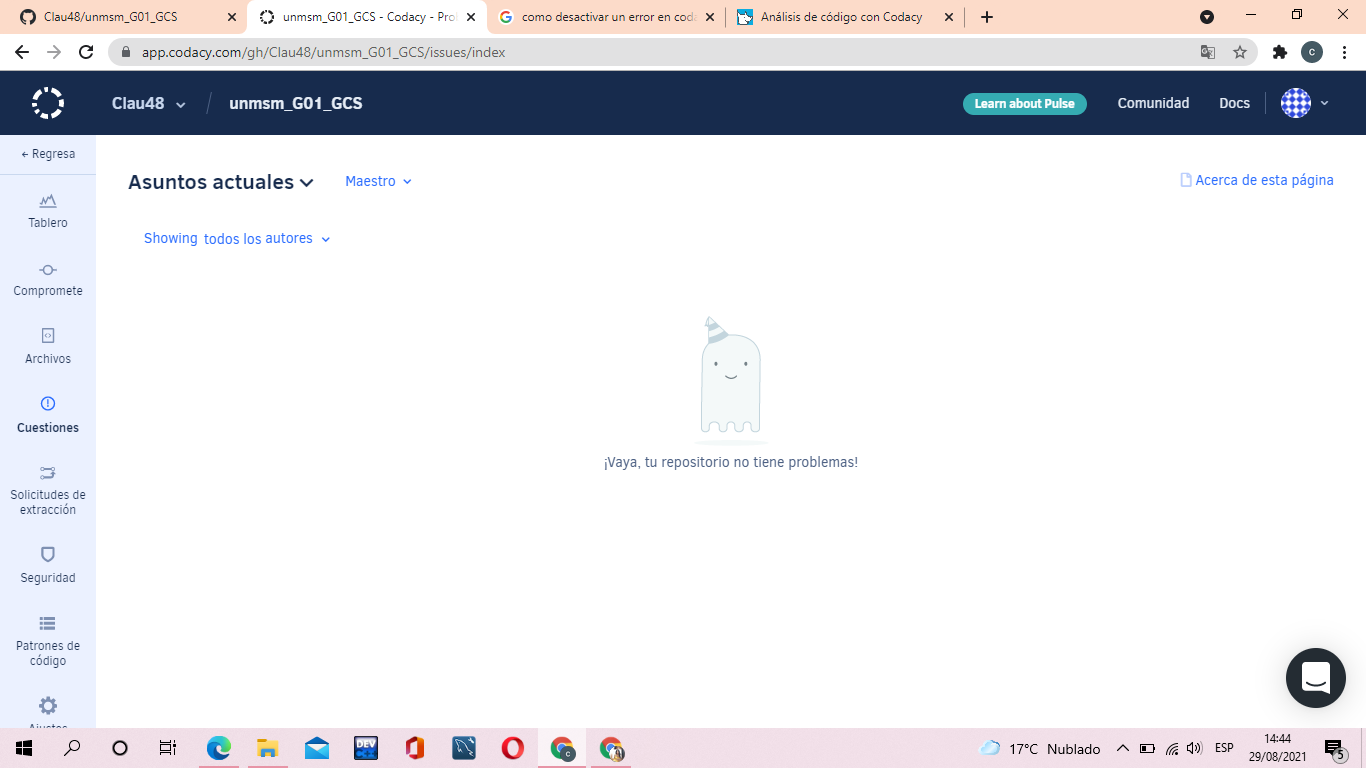
# Code Patterns

Cada code Patterns correctamente hecho nos muestra un check de que nuestro código está correcto.



# Issues

Las cuestiones fueron corregidas en el proyecto, por lo que muestra que no tenemos errores en nuestro programa, dándonos así una calificación A en la calidad de código.



# Archivos

Cada archivo del proyecto de la página web MM&P tiene una calificación de A ya que sus errores han sido corregidos respectivamente por cada uno de los integrantes del proyecto.

