**Práctica Final**

Contenido

[Práctica 2 2](#_Toc104576450)

[1. Define el POS asociado al mismo. 2](#_Toc104576451)

[2. Crea el RBS del proyecto con un mínimo de 3 o 4 niveles de detalle. 3](#_Toc104576452)

[Práctica 5 3](#_Toc104576453)

[1. Crea el WBS hasta un nivel 2-3. ¿Usarías el RBS como base o encuentras alguna 3](#_Toc104576454)

[alternativa más sencilla y/o clara? 3](#_Toc104576455)

[2. Con el primer nivel del WBS (suponiendo que son actividades) 4](#_Toc104576456)

[3. Representación en diagramas de las actividades (primer nivel) y sus relaciones (estimaciones en %) 5](#_Toc104576457)

[4. Crea un presupuesto a partir del total que aparece en el pliego (resta impuestos y el margen comercial de un 20%). 6](#_Toc104576458)

[Práctica 6 7](#_Toc104576459)

[1. Con la estimación de costes de la práctica anterior dimensiona el equipo e identifica los roles que son necesarios para este proyecto. Ajusta retribución que has obtenido con la media del mercado. 7](#_Toc104576460)

[Práctica 9 8](#_Toc104576461)

[1. Gestión de la configuración 8](#_Toc104576462)

[3. Gestión de la integración continua (sin docker) 14](#_Toc104576463)

[Práctica 10 17](#_Toc104576464)

[1. Considerando el pliego 17](#_Toc104576465)

[2. Considerando el último entregable incluido en la práctica 9 19](#_Toc104576466)

[Práctica 11 21](#_Toc104576467)

[1. Considerando el pliego 21](#_Toc104576468)

# Práctica 2

A partir del pliego

## Define el POS asociado al mismo.

* 1. Problem/Opportunity

Diversos servicios municipales de Castellón no pueden ofrecer amplia información al ciudadano de manera fácil, rápida y actualizada.

* 1. Goals

Los ciudadanos de Castellón podrán acceder a multitud de servicios para poder acceder a información de diferentes ámbitos publicada por el ayuntamiento. Servicios como eventos, transporte público, trámites y gestiones, etc.

* 1. Objectives

Desarrollar una aplicación móvil (Android e iOS) para conseguir mejorar la difusión de información importante en la ciudad de Castellón, en beneficio a los vecinos y visitantes de la ciudad.

* 1. Success Criteria
     + Cantidad de descargas
     + Cantidad de usuarios
     + Cantidad de reseñas positivas
  2. Assumptions, Risks, Obstacles
     + Incumplimiento de los requisitos.
     + Tiempo/Recursos/Personal

## Crea el RBS del proyecto con un mínimo de 3 o 4 niveles de detalle.

# Práctica 5

A partir del pliego

## 1. Crea el WBS hasta un nivel 2-3. ¿Usarías el RBS como base o encuentras alguna

## alternativa más sencilla y/o clara?

## 2. Con el primer nivel del WBS (suponiendo que son actividades)

1. **Define entregas (Milestones). Agrupa actividades.**
   * Hitos:
     1. Análisis de requisitos y diseño detallado.
     2. Desarrollo de la solución.
     3. Instalación, Configuración y Parametrización.
     4. Puesta en Marcha y Formación.

* Agrupación:

Las dos actividades, “crear la aplicación” y “crear el servidor web” pertenecen a los 4 hitos.

1. **Asigna una estimación de horas a cada actividad considerando los plazos totales del pliego**

La prestación iniciará su ejecución a partir del día siguiente a la formalización del contrato, y dicha ejecución finalizará como máximo en un plazo de **10 meses**.

Estimación de horas:

* Hacer la aplicación (IOS y Android): **7 meses**
* Hacer el servidor web: **3 meses**

1. **Identifica interconexiones/dependencias entre actividades**

La actividad “*hacer la aplicación*” tiene como dependencia el mostrar la agenda y los eventos de la web en la aplicación, es decir, si la actividad “*hacer el servidor web*”no está completada, la primera actividad no podrá mostrar el contenido de la segunda actividad.

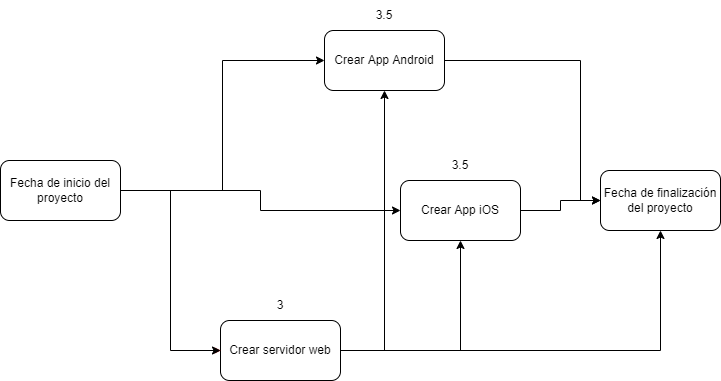
## 3. Representación en diagramas de las actividades (primer nivel) y sus relaciones (estimaciones en %)

1. **Crea un Gantt**
2. **Crea un Pert**

Los números del diagrama de Pert es el tiempo medido en meses.

Podemos observar que para finalizar el proyecto hay que tener la aplicación (iOS y Android) creada y el servidor web. Pero para poder terminar las aplicaciones necesitamos terminado el servidor web para poder mostrar la agenda y eventos en la aplicación (iOS y Android).

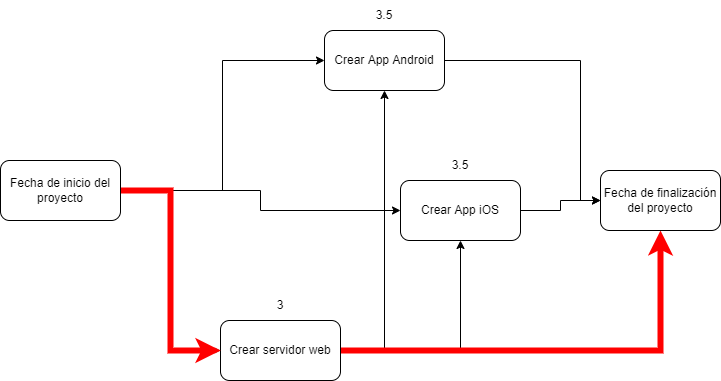
Se puede ver como las actividades se pueden empezar a realizar sin depender una de la otra, pero para terminarla está la pequeña dependencia comentada anteriormente.

****

1. **Busca el camino crítico**

El camino crítico está indicado con flechas rojas.

Para poder finalizar el proyecto el servidor web tiene que estar creado, por lo tanto, si la actividad de crear el servidor web no está completado, las actividades de crear la aplicación tanto en Android como en iOS no podrán completarse, por lo tanto, no podría completarse ninguna actividad.



## 4. Crea un presupuesto a partir del total que aparece en el pliego (resta impuestos y el margen comercial de un 20%).

El importe máximo de licitación se establece en 82.500€ más IVA.

Como hay que restar los impuestos el presupuesto empieza desde 82.500€

El margen comercial corresponde un 20%, lo que supone -> 82.500 x 0.20 = 16.500€.

Por lo tanto, el presupuesto seguirá la siguiente fórmula:

**Base de licitación – Margen comercial = 82.500€ - 16.500€ = 66.000€**

# Práctica 6

A partir del pliego

## 1. Con la estimación de costes de la práctica anterior dimensiona el equipo e identifica los roles que son necesarios para este proyecto. Ajusta retribución que has obtenido con la media del mercado.

Lo roles que son necesarios para este proyecto son:

1. Un **responsable del proyecto** que tendrá dedicación parcial al proyecto y que se identificará como el principal interlocutor con el Ayuntamiento de Castellón. Coordinará y garantizará la adecuada cobertura del servicio.
2. Un **perfil funcional** con experiencia demostrable en los puntos descritos en el apartado 3 del presente documento con dedicación exclusiva para las tareas del proyecto.
3. **Equipo de trabajo base**, con experiencia técnica demostrable en los puntos descritos en el apartado 3 del presente documento con dedicación exclusiva para las tareas del proyecto.

En cuanto a la retribución, el salario medio de los profesionales IT es de 37.660€ brutos al año. El proyecto son 10 meses, por lo tanto, serían 31.334€ por cada perfil tecnológico.

La dimensión del equipo será **1** responsable del proyecto, **1** perfil funcional y **1** persona como equipo de trabajo base. No he ampliado el número de personas en el equipo de trabajo base debido al presupuesto.

Al ser tres perfiles tecnológicos, el precio total a pagar entre los tres sería de 31.334 x 3 = 94.002. Debido a que no es una aplicación tan sofisticada e importante, se les reducirá el salario.

Por lo tanto, al tener como presupuesto 66.000€, el precio a pagar a cada miembro será:

* Responsable de proyecto: 24.000€
* Perfil funcional: 22.000€
* Equipo de trabajo base: 20.000€

# Práctica 9

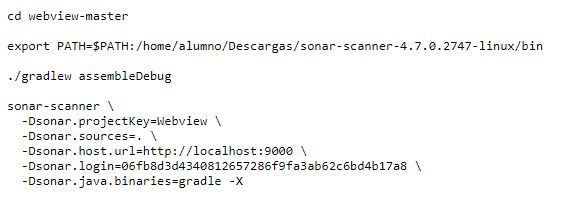
## 1. Gestión de la configuración

1. **Crea un repositorio público (o usa uno que ya tengas) en Github u otro servicio similar.**

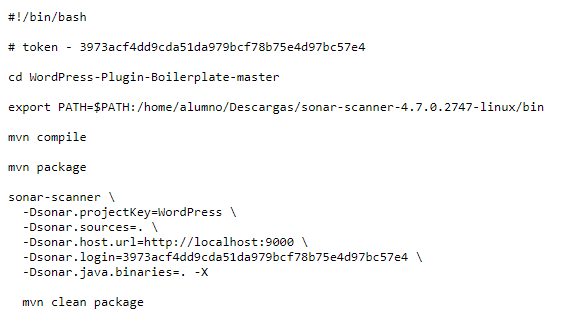
<https://github.com/alvaritocaro/GPI-2.git>

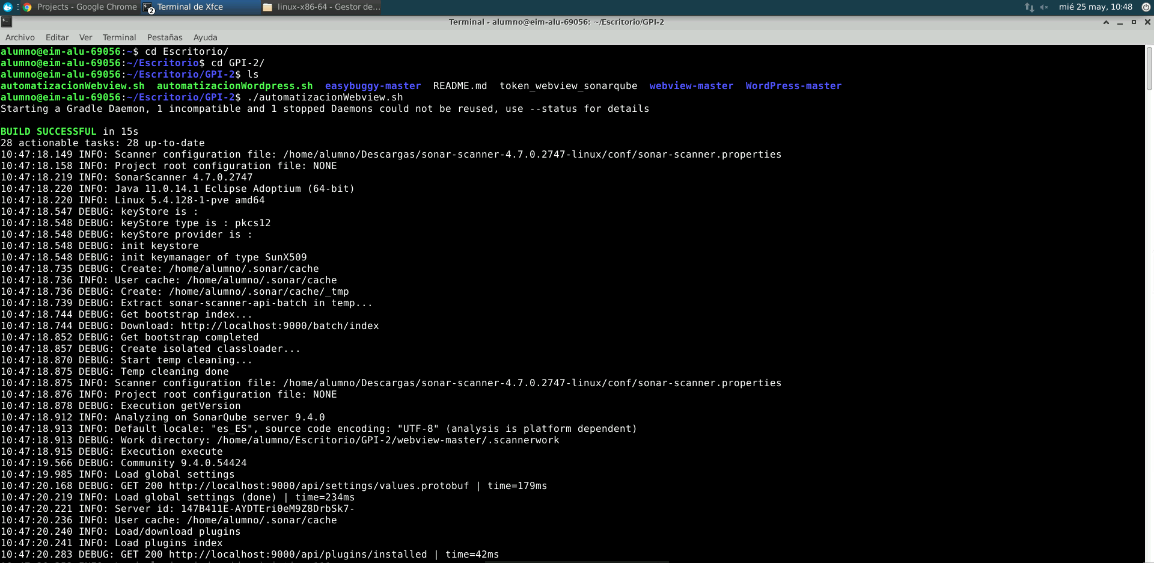
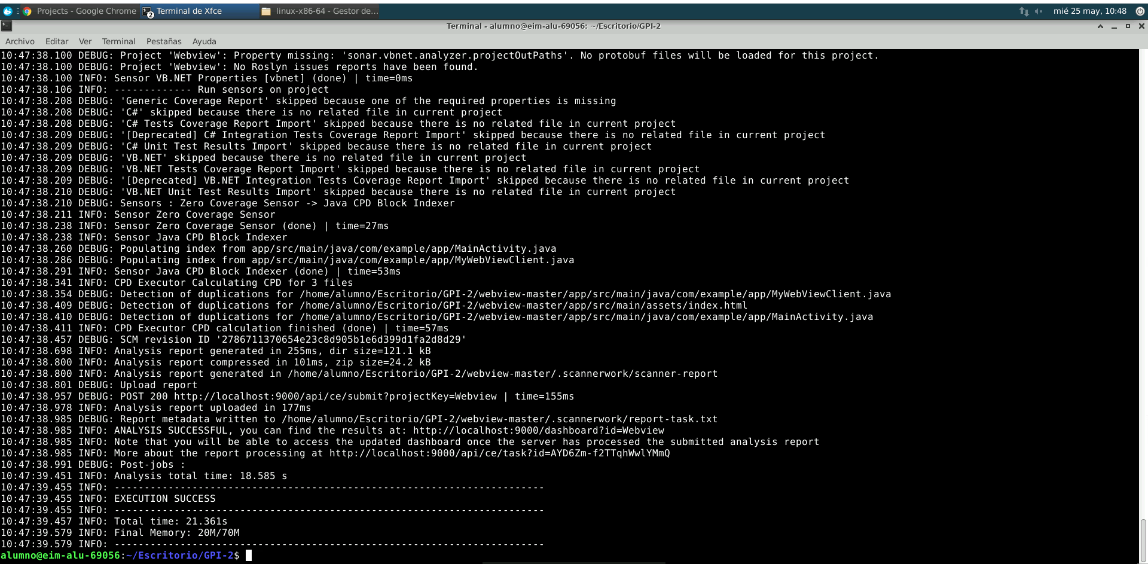
1. **Organiza los entregables descritos en práctica 7 junto con la automatización para compilar, empaquetar, el análisis estático (se puede usar cualquier librería de análisis estático, no es obligatorio integrar sonarqube) y otro script para ejecutar las pruebas.**

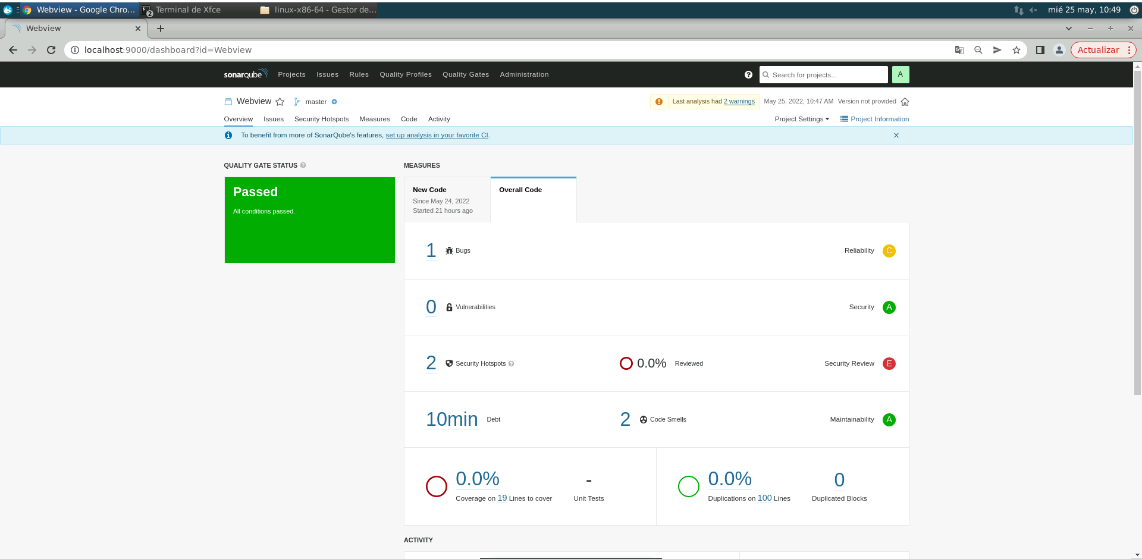
* **Webview:**

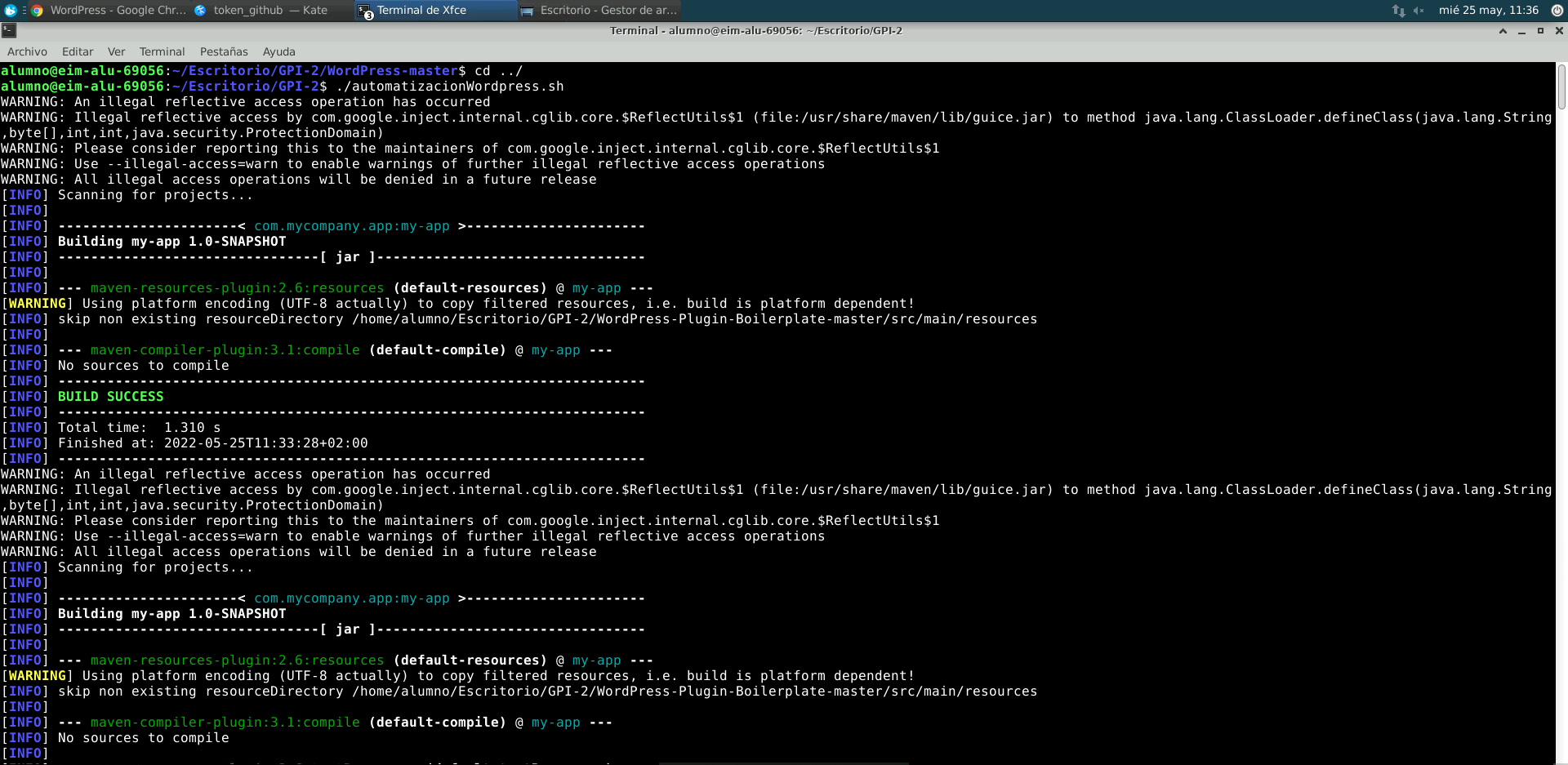


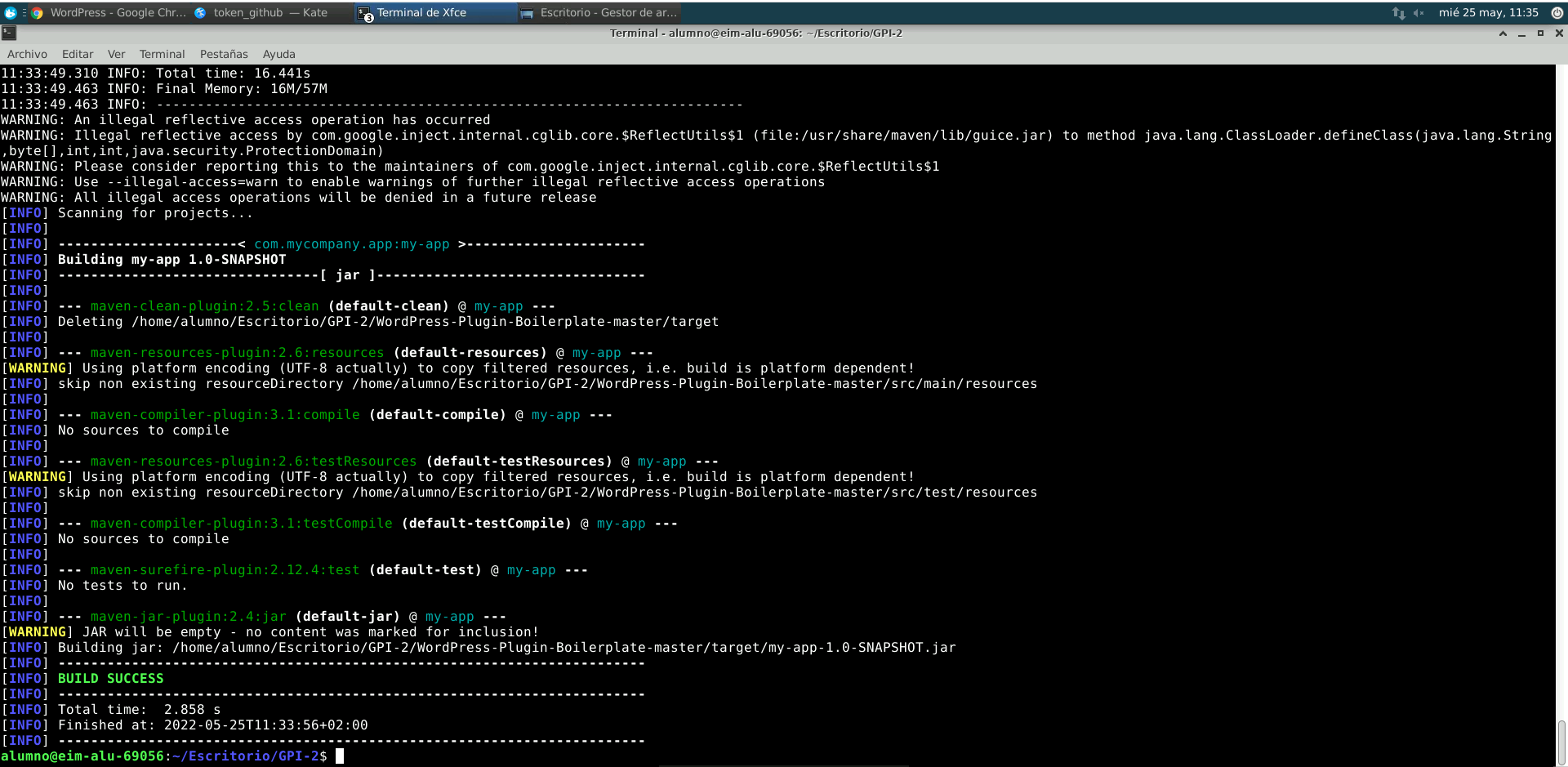
* **WordPress:**

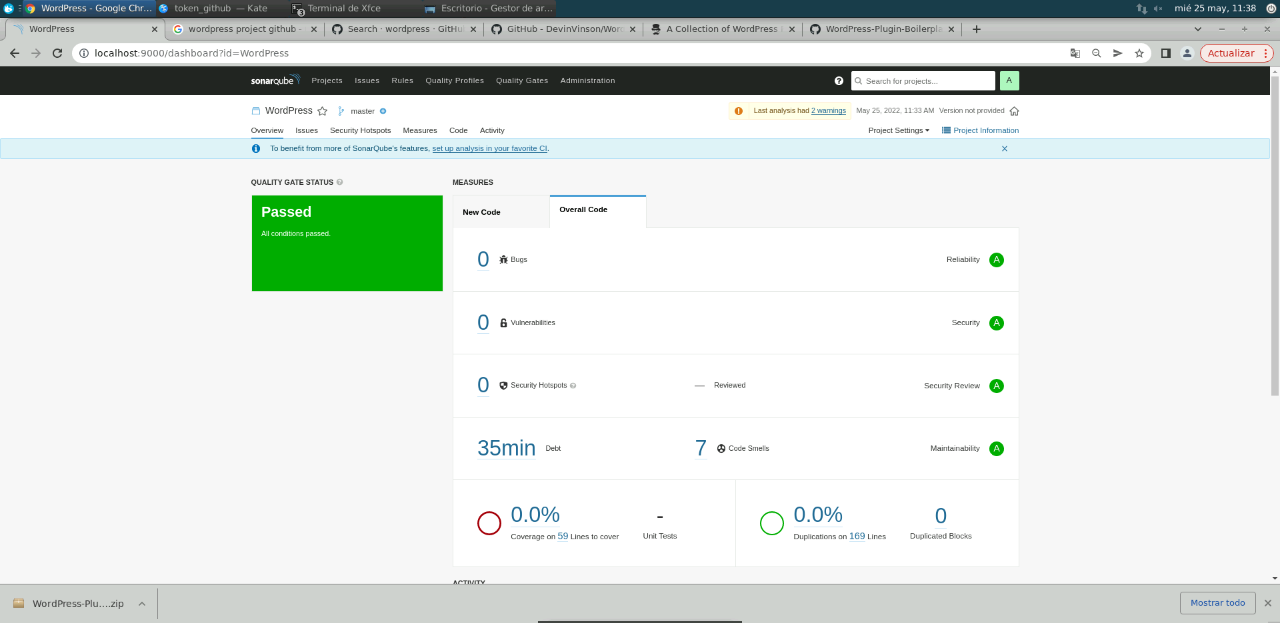
****

1. **Ejecuta el proceso completo en el contenedor de la asignatura (descarga de git+ejecución del script) y obtén evidencias de que el proceso ha funcionado correctamente.**
   * **Webview:**

****

* + **WordPress:**

****

****

1. **Define en un gráfico los tags/branches adaptando el diagrama siguiente (reléase line) para cumplir los entregables previstos en el pliego (no hace falta implementarlos en vuestro repositorio). Puedes tener uno con los entregables o un repositorio para cada entregable.**

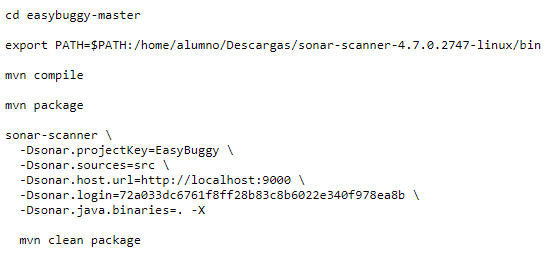
El diagrama (reléase line) es perfecto. De este modo separas el mantenimiento y el desarrollo en líneas de código distintas.

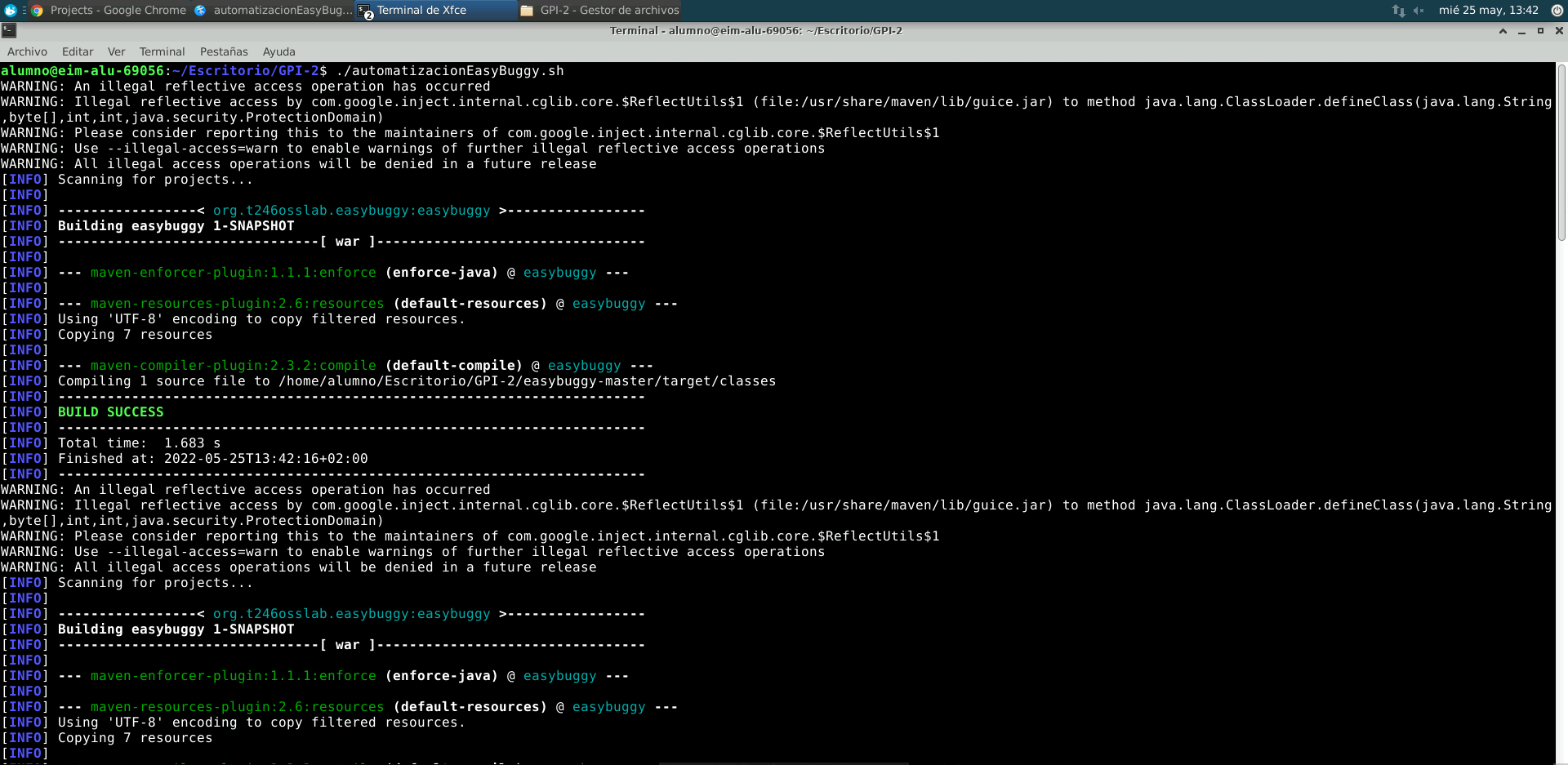
**Diagrama, Esquemático

Descripción generada automáticamente**

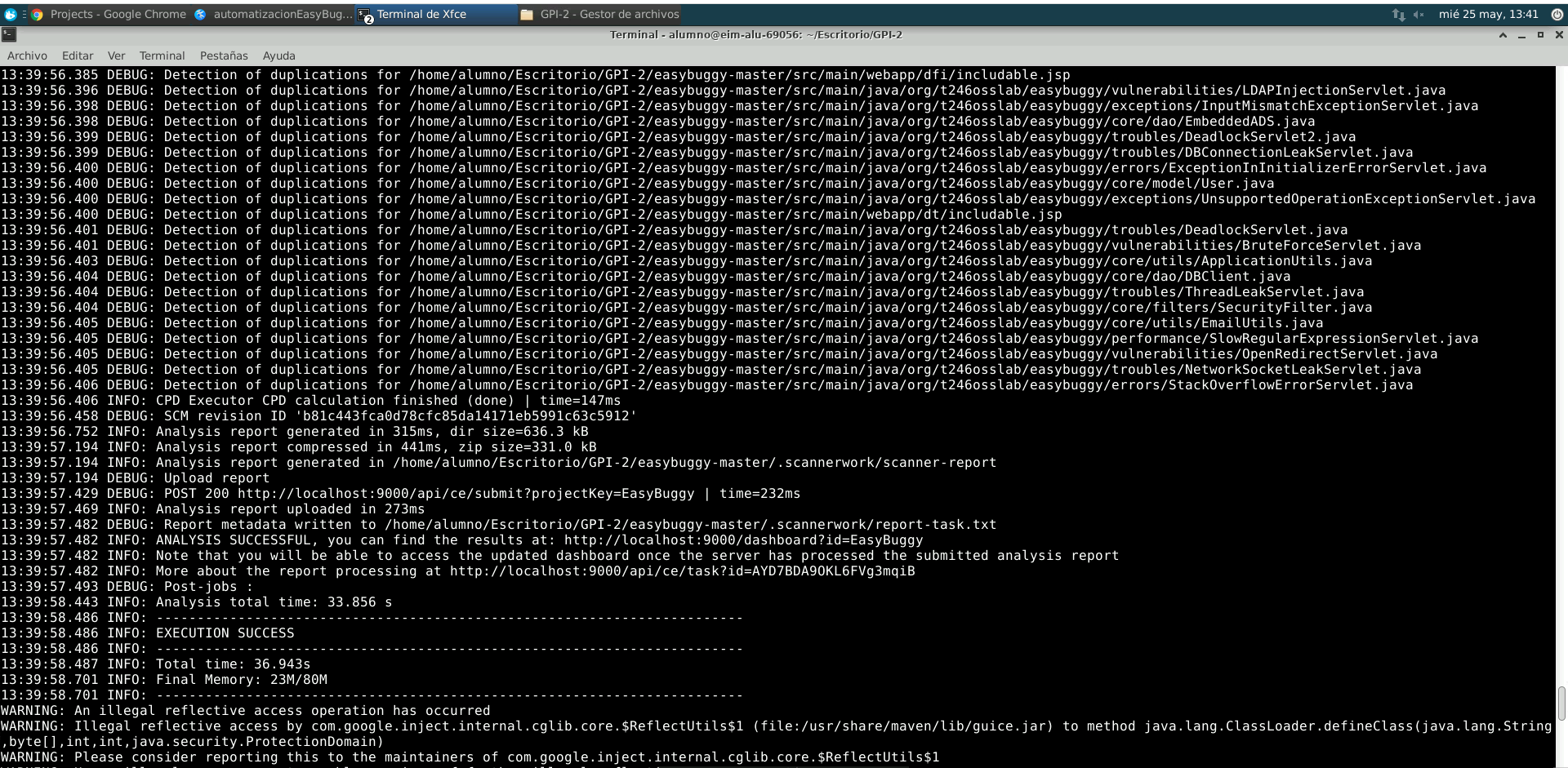
1. **Incorpora un nuevo entregable en el mismo repositorio o en uno independiente, según lo que se considere más cómodo para su evolución: el proyecto a integrar es easybuggy en tu repositorio e inclúyelo en tus scripts la compilación:**

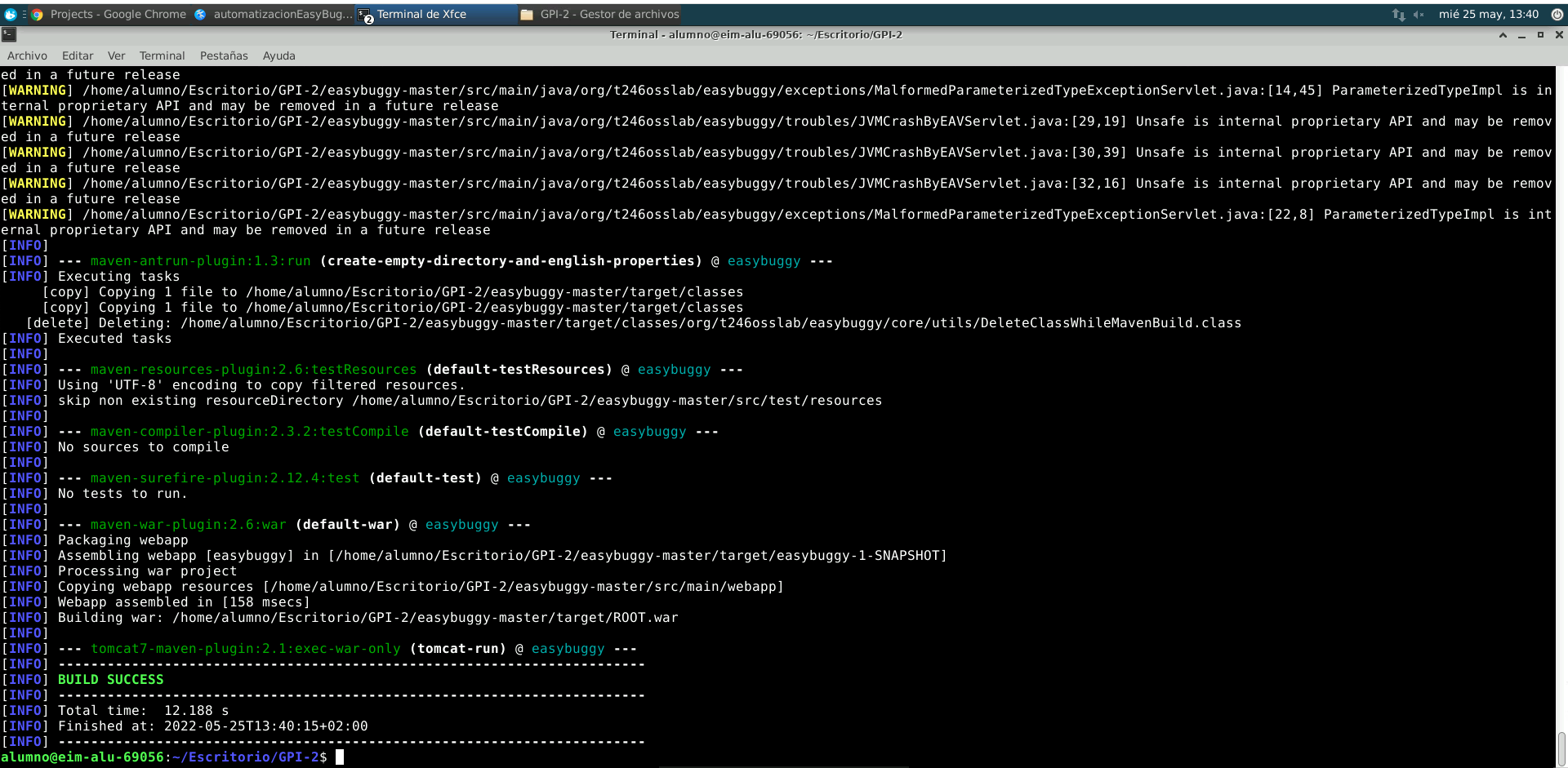
<https://github.com/k-tamura/easybuggy>











## Gestión de la integración continua (sin docker)

1. **Instala Jenkins en el contenedor de la asignatura como una webapp. Para ello hay que disponer de un servidor de aplicaciones Java (p.ej Tomcat). Comprobad antes si en el contenedor está instalado Tomcat.**

Texto

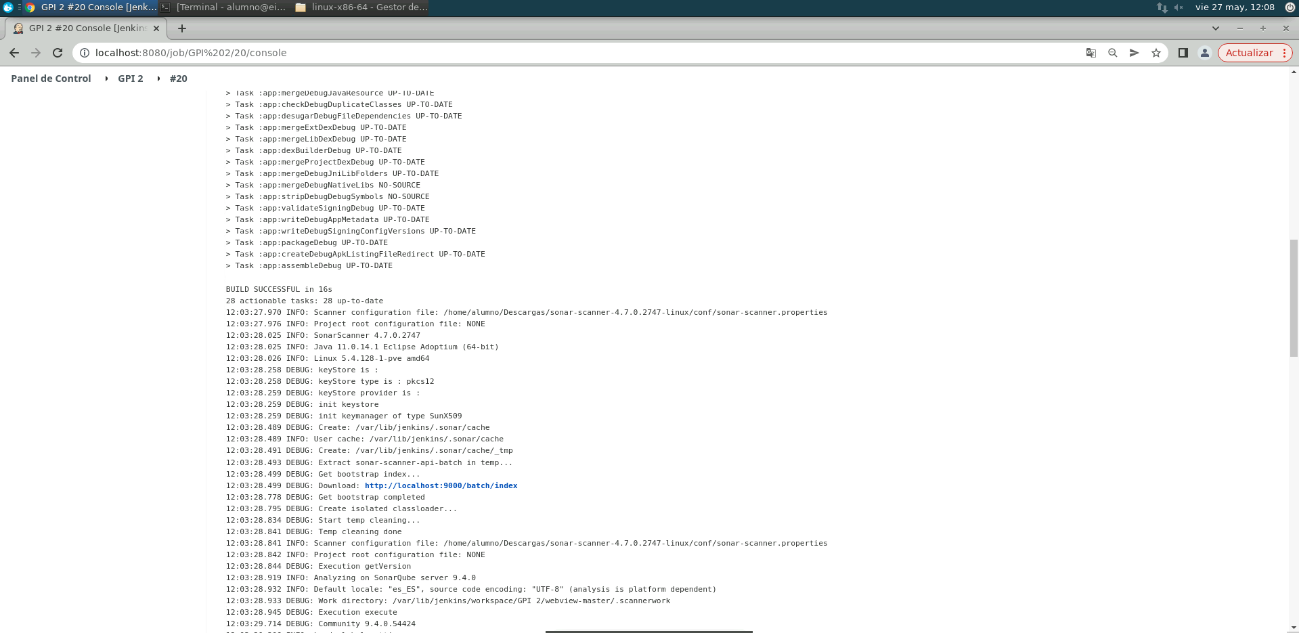
Descripción generada automáticamente

1. **Configura un workflow con mínimo dos bloques (stages), compilación/empaquetado/análisis y pruebas).**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteHe ejecutado el pipeline, pero se me queda atascado todo el rato el contendor y no finaliza correctamente la ejecución.

Interfaz de usuario gráfica, Texto, Aplicación

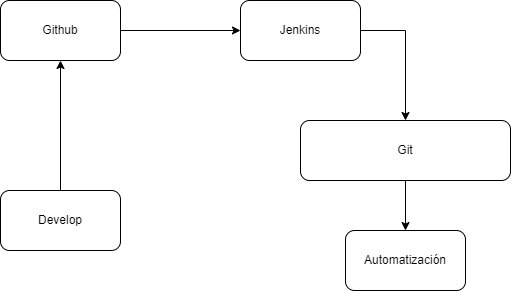
Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

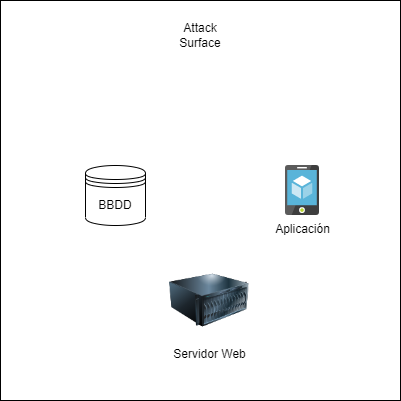
Descripción generada automáticamente

1. **Realiza un diagrama de tu sistema CI similar al que se adjunta. Incluye urls de github, rutas del filesystem del contenedor, etc.**



# Práctica 10

## 1. Considerando el pliego

1. **Identifica un mínimo de 5 casos de abuso.**
   1. Cambiar coordenadas para recibir diferentes eventos
   2. Cambiar fecha del móvil para apuntarse a eventos anteriores
   3. Mandar correo haciéndose pasar por el correo del Ayuntamiento de Castellón
2. **Identifica el attack surface de los componentes que componen la solución.**

La superficie de ataque del entorno software es: la **base de datos**, el **servidor web** y la **aplicación** (Android e IOS).

1. **Emplea el modelo STRIDE e identifica en cada categoría una amenaza particularizada al sistema considerado**.

S -> Spoofing -> Como a la aplicación se puede acceder como usuario y no tiene doble factor pueden hacer shoulder surfing y robarte las credenciales.

T -> Tampering -> Como a la aplicación se puede acceder como usuario puede realizar un ataque SQL Injection para robar datos.

R -> Repudation -> La aplicación no tiene un sistema de logs donde el usuario quede registrado por lo que si realiza un ataque no dejará rastro.

I -> Information Disclosure -> El usuario produce un error donde hace que se muestra información confidencial.

D -> Denial of service -> Se puede realizar un ataque DoS y que la web sea inaccesible.

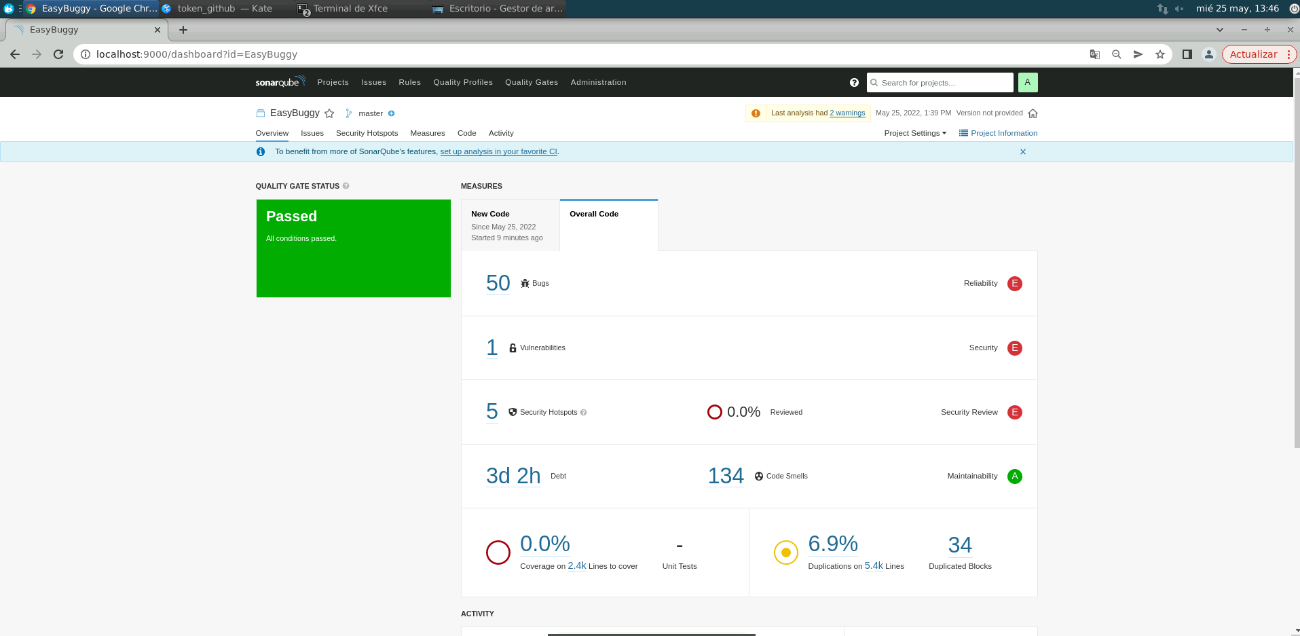
E -> Elevation of privileges -> El usuario consigue permisos de administrador en la web y puede modificar el contenido.

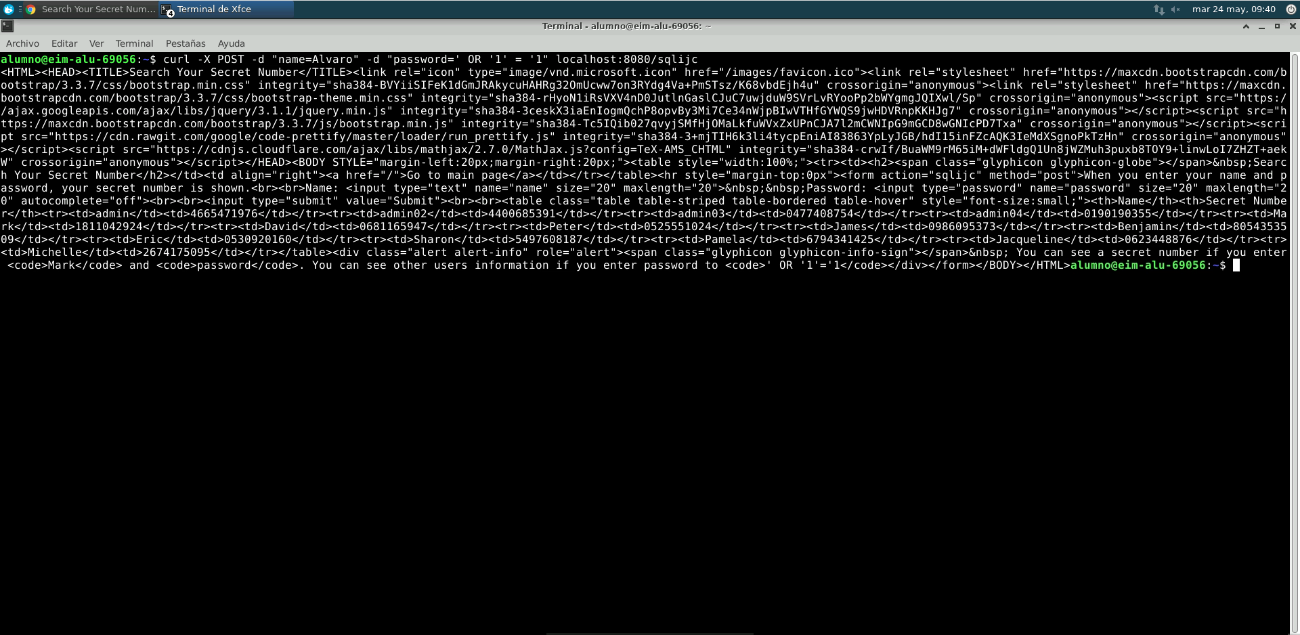
1. **Lista un mínimo de 5 tecnologías de seguridad aplicables según la arquitectura del sistema considerado.**
   1. Inicio de Sesión Único (SSO)
   2. Firewall de aplicación
   3. Cifrado
   4. Logging
   5. Cliente VPN para acceder a la administración del CMS
2. **Lista las 3 últimas vulnerabilidades de alguno de los componentes de terceros integrados en la solución y describe cómo afectarían cada una de esas vulnerabilidades al sistema. ¿Alguna vulnerabilidad requiere de un parcheado urgente? ¿Por qué?**
   1. **Omisión de autenticación (Authentication bypass):** Permitiría a un atacante saltarse los requisitos de autenticación en el login y acceder a secciones que solo un usuario registrado puede acceder.
   2. **Vulnerabilidad de inyección de objetos PHP**: Un usuario puede enviar un código no desinfectado, es decir, un PHP malicioso a la página.
   3. **Vulnerabilidad de puerta trasera**: Permite a los usuarios no autorizados eludir las medidas de seguridad y obtener acceso de alto nivel al servidor o a la página web.

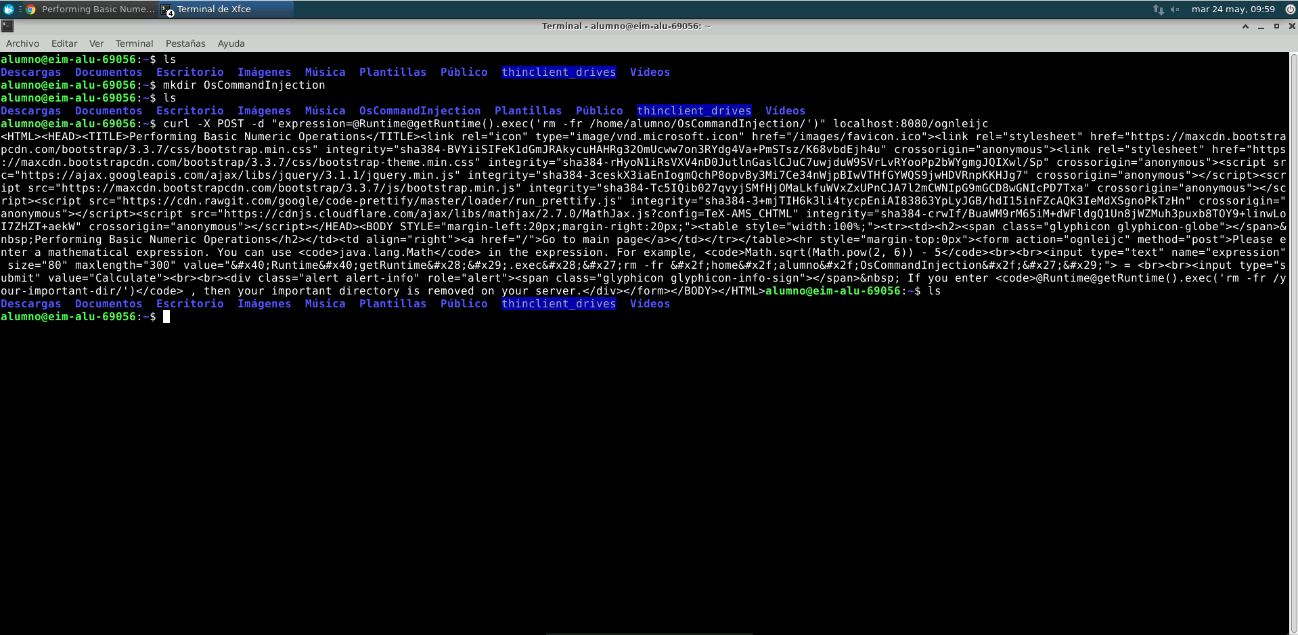
Sí, la tercera vulnerabilidad requiere un parcheado urgente ya que tener acceso de alto nivel al servidor o a la página web puede causar graves problemas.

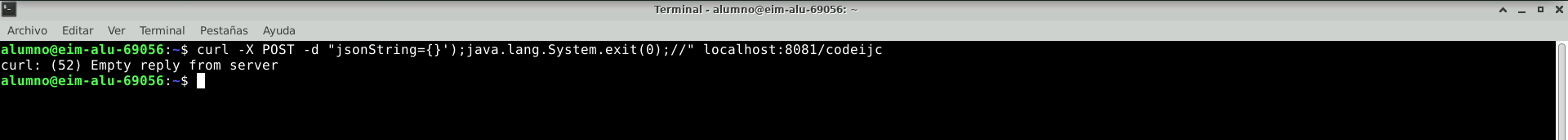
## 2. Considerando el último entregable incluido en la práctica 9

https://github.com/k-tamura/easybuggy

1. **Incluye en el análisis de código (recomendado Sonarqube) este nuevo entregable y comprueba el resultado del análisis estático.**
2. De las TOP 10 owasp identifica 3 de ellas en la webapp easybuggy
   1. **Utiliza curl para automatizar, ejecuta en el contenedor para comprobar las 3 y obtén evidencias de ellas.**
      1. SQL Injection:

**curl -X POST -d “name=Alvaro” -d “password= ‘ OR ‘1’ = ‘1” localhost:8080/sqlijc**

* + 1. Os Command Injection:
  + **mkdir OsCommandInjection**
  + **curl –X POST –d “expression=@Runtime@getRuntime().exec(‘rm -fr /home/alumno/CodeInjection’)” localhost:8080/ognleijc**
    1. Code Injection:
       - Texto

         Descripción generada automáticamente**Curl -X POST -d “jsonString={}’);java.lang.System.exit(0);//” localhost:8081/codeijc**
  1. **Selecciona un escáner web que se pueda instalar y usar en el contenedor y obtén evidencia de lo que detecta.**

# Práctica 11

## 1. Considerando el pliego

1. **Seleccionad los requisitos del pliego que requieren datos personales y qué nivel de seguridad requieren**.

* Cuando el usuario utiliza esta comunicación con el servicio 010 se verá involucrado el número de teléfono. Por lo tanto, será un nivel de seguridad básico.
* Cuando el usuario se registra se requieren datos personales de un nivel de seguridad básico.
* Cuando el usuario hace algún trámite o gestión se requieren datos personales con un nivel de seguridad medio.

1. **Seleccionad los requisitos del pliego que requieren cookies y de qué tipo son (opt-in o opt-out**).

* La selección del idioma mediante las llamadas “*Cookies de sesión”*. (opt-in).
* La ubicación mediante las llamadas “*Cookies de seguimiento*”.
* Las redes sociales que vinculan con la web mediante las llamadas “*Cookies de redes sociales*”.
* Las que aportan información sobre el comportamiento de los usuarios en la web mediante las llamadas “*Cookies de análisis*”.
* Revisad obligaciones según el licenciamiento de los proyectos que hemos integrado:
  1. **Incluid la licencia/s originales del creador/es en un apartado de vuestro github e incluid el/los enlace/es en la práctica.**

Las licencias se encuentran en una carpeta de github llamada “Licencias” donde encontrarás las licencias de los proyectos EasyBuggy y WordPress.

El proyecto de webview no contiene licencia.

Webview:

Enlace creador oficial: <https://github.com/slymax/webview.git>

WordPress:

Enlace creador oficial: <https://github.com/DevinVinson/WordPress-Plugin-Boilerplate.git>

EasyBuggy:

Enlace creador oficial: <https://github.com/k-tamura/easybuggy.git>

* 1. **¿Podemos redistribuirlo? ¿Y cobrar por ello?**

EasyBuggy:

En el punto 4 de la licencia, señala que se puede redistribuir y cobrar por ello.

WordPress:

En la línea 296 de la licencia, señala que se puede redistribuir. Se puede cobrar por ello.

Webview:

Al no tener licencia en github se aplica las leyes de derecho de autor predeterminadas, lo que implica que conservas todos los derechos de tu código fuente, y nadie puede reproducir, distribuir o crear trabajos a partir de tu trabajo.