# **SECURE SOFTWARE DEVELOPMENT**

**(GIIF)**



***Unai Fernández Beñat Izagirre Aitor Irusta Oier García***

Secure Software Development

20/12/2021

Grupo GIIF.inc

**ÍNDICE**

Contenido

[**SECURE SOFTWARE DEVELOPMENT** 1](#_heading=h.gjdgxs)

[Introducción 4](#_heading=h.30j0zll)

[Objetivos 4](#_heading=h.1fob9te)

[Requisitos 4](#_heading=h.3znysh7)

[Arquitectura 6](#_heading=h.2et92p0)

[Plano lógico de nuestra arquitectura 6](#_heading=h.tyjcwt)

[Especificaciones 6](#_heading=h.3dy6vkm)

[Funcionalidad 7](#_heading=h.1t3h5sf)

[Aplicación WEB 7](#_heading=h.4d34og8)

[Aplicación Android 17](#_heading=h.17dp8vu)

[Seguridad 24](#_heading=h.3rdcrjn)

[Ataques de reconocimiento 24](#_heading=h.26in1rg)

[Configuración del apache y WordPress 25](#_heading=h.lnxbz9)

[Vulnerabilidades OWASP 29](#_heading=h.35nkun2)

[Vulnerabilidades API REST 33](#_heading=h.1ksv4uv)

[Conclusiones 33](#_heading=h.44sinio)

[Líneas futuras 34](#_heading=h.2jxsxqh)

[Referencias 34](#_heading=h.z337ya)

[Anexos 34](#_heading=h.3j2qqm3)

# Introducción

El presente documento es la memoria técnica presentada por los alumnos, Aitor Irusta, Unai Fernández, Oier García y Beñat Izagirre del proyecto “Secure Software Development”. Este reto forma parte del curso de Lanbide “Desarrollo de aplicaciones para dispositivos Android” impartido en el centro Maristak de Durango con la metodología ETHAZI. La duración de este reto es de 380 horas repartidas entre el 16/09/2021 y el 22/12/2021. El reto consiste en desarrollar una aplicación web segura y una aplicación móvil segura. Para la creación de estas, hemos usado herramientas específicas las cuales mencionaremos en el documento presente.

# Objetivos

El objetivo de este es el desarrollo de una nueva aplicación que sea multiplataforma y se adecue a las necesidades de movilidad de los empleados. Esta aplicación contiene una serie de especificaciones sobre la funcionalidad de la aplicación y la seguridad de esta.

Otro objetivo que tenemos en este proyecto más allá del resultado del propio proyecto es el aprendizaje de desarrollo seguro tanto web como Android.

# Requisitos

El reto a realizar venía con unos requisitos y metas a llegar:

* Los empleados tienen que poder registrar los gastos realizados en los viajes que realiza en nombre de la empresa.
* La aplicación tendrá que contar con un mecanismo de autenticación y de autorización.
* Gestionar las tarjetas de crédito europea o internacional para que por políticas de la empresa no puedan estar ambas activadas al mismo tiempo. Para poder realizar esto necesitaremos implementar en nuestras aplicaciones una comunicación con una Web Services del Banco, respetando sus especificaciones técnicas y de seguridad (especificadas en un documento técnico facilitado al principio del reto).
* Securizar la aplicación aplicando herramientas de seguridad basadas en la normativa vigente, para la protección de la información y elaborar un informe del procedimiento que se ha seguido donde se definen los requisitos de seguridad implantados siguiendo las recomendaciones OWASP.
* Deberá de tener una navegación basada en un menú, lista o similar permite seleccionar una opción y mostrar su detalle.
* Diferentes roles dentro de la aplicación. Habrá roles de usuario y administrador y depende de los roles se podrán acceder a funcionalidades relacionadas con un empleado común o funcionalidades relacionadas con administradores.
* Deberá de tener unos apartados de “ajuste” y “acerca de” que muestre información sobre la versión de la aplicación.
* Debe de tener una barra de navegación en la barra inferior.
* Deberá de incluir un icono con una imagen del usuario.
* Deberá de incluir una sección de perfil donde el usuario podrá ver la información de su propia cuenta (nombre y apellidos, DNI, nombre de usuario y última conexión).
* Tendrá que haber una sección para introducir los gastos producidos para la aprobación de estos. Este apartado cuenta con sus propias especificaciones:
  + Calcular el precio de los kilómetros recorridos en caso de que se haya usado transporte propio.
  + En caso de que un empleado esté menos de 4 días fuera, se le permitirá solicitar dietas sin introducir gastos.
  + Las dietas serán de 60€/día en Europa y 100€/día fuera de Europa.
  + La aplicación deberá calcular el total de gastos.
  + El gasto se podrá imputar a un proyecto o a un departamento
* Tiene que haber una sección para que los empleados puedan registrar las horas trabajadas diariamente en cada proyecto y controlar que un empleado no pueda trabajar más de 40 horas semanales.
* También habrá un apartado para ver los gastos que se han realizado con la opción de filtrarlos según las fechas y según el costo del gasto.

# Arquitectura

## Plano lógico de nuestra arquitectura

Diagrama

Descripción generada automáticamente

## Especificaciones

### Aplicación WEB

* El servidor está alojado en VMware ESXi y está usando el sistema operativo Debian 10. La dirección IP del servidor es 10.122.27.95.
  + El hipervisor VMware ESXi aparte de brindarnos ventajas en cuanto al rendimiento de la máquina y la gestión centralizada, también nos ofrece la ventaja de crear instantáneas de la máquina.
  + Debian se caracteriza por parchear vulnerabilidades rápidamente.
  + Es un sistema operativo muy estable y suele dar pocos errores y fallos.
* El servidor web está usando apache2, en la versión 2.4.38.
  + Es código abierto y gratuito
  + Ofrece una gran comunidad que otorga soporte frente a vulnerabilidades.
  + Gran escalabilidad y flexibilidad
* El sistema gestor de base de datos que estamos usando es MariaDB 10.3.31.
  + Eficiencia en consultas complejas
  + Compatibilidad con la sincronización con SQLite
* El CMS que estamos usando es WordPress 5.8.2.
  + Es gratis y de código abierto.
  + Extremadamente flexible y versátil.
* La versión de PHP que estamos usando es el 7.3

### Aplicación Android

* Estamos desarrollando la aplicación Android usando Android Studio 2020.3.
  + Ejecución de la app en tiempo real gracias al emulador.
  + Tiene renderizado en tiempo real, layouts y puede hacer uso de parámetros tools.
* La versión de Java que está usando el sistema gestor es el 11.0.10.
* El sistema gestor de base de datos de la aplicación Android es SQLite 3.35.5.
  + SQLite es un proyecto de código abierto disponible en dominio público.
  + Adecuado para el desarrollo básico y pruebas.
  + Compatible con Android.
  + Compatible con sincronización con MariaDB.
* Los móviles que estamos virtualizando para desarrollar el programa son PIXEL 4A y PIXEL 5 ambos usan la API 30 y Android 8.

### Otros recursos

* Estamos usando PuTTY SSH para conectarnos vía SSH al servidor Debian.
* Estamos usando Eclipse IDE for Java Developers para testear en local.
* Usamos el DB browser (SQLite) para ver de forma gráfica la base de datos SQLite.
* Usamos MySQL Workbench para conectarnos de forma remota a la base de datos de WordPress.
* Google Drive y GitHub para guardar versiones, compartir proyectos y hacer copias de seguridad.

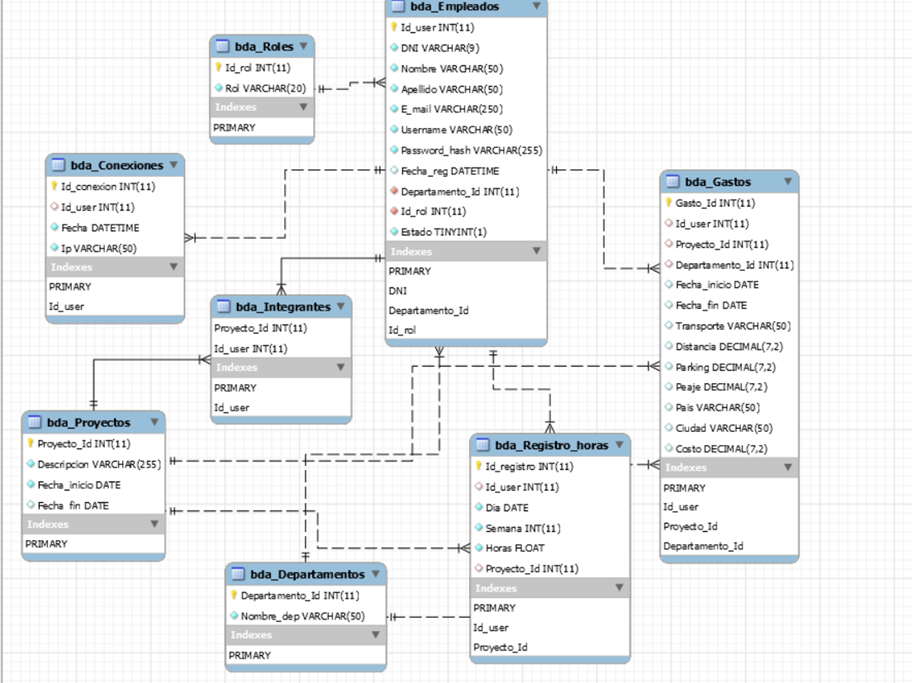
# Funcionalidad

## Aplicación WEB

Hemos desarrollado una aplicación web usando WordPress para que los usuarios puedan registrar los gastos, las horas trabajadas y activar la tarjeta bancaria. Además, los administradores podrán gestionar los diferentes apartados (usuarios, proyectos, departamentos) y además podrá los gastos registrados por los empleados

### Base de datos

Para que todo esto funcione hemos tenido que desarrollar una base de datos en MariaDB. El esquema relacional de la base de datos seria esta:



El resumen de cada una de las tablas sería esta:

bda\_Empleados:

Guardar los datos de los usuarios (DNI, nombre, apellido etc.) y relacionarlo con varias claves foráneas (departamentos, rol …).

bda\_Roles:

Establecer los roles, de momento solo usamos 2, usuario y administrador.

bda\_Conexiones:

Un histórico de los inicios de sesión realizados, junto con la fecha y la IP por si esta intentado acceder desde una IP extraña.

bda\_Departamentos:

Establecer los departamentos donde trabajan los empleados.

bda\_Registro\_Horas:

Registrar las horas que trabajan los empleados.

bda\_Gastos:

Registrar los gastos insertados por los empleados.

bda\_Integrantes:

Establecer en que proyectos trabajan los empleados.

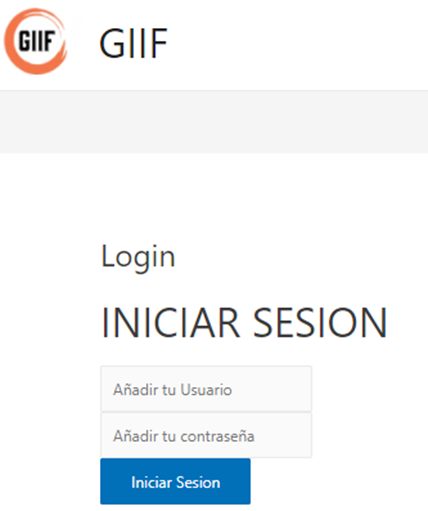
bda\_Proyectos:

Establecer los proyectos que están en la empresa.

### Navegación

#### Zona login y perfil

El usuario o administrador hará login en <https://10.122.27.95/> donde le saldrá nuestra página de login. A nivel de base de datos, los usuarios están siendo guardados en una tabla propia en vez de una tabla creada por WordPress. Estos nos ofrecen más manejabilidad a la hora de crear campos nuestros en la tabla de usuarios que necesitamos usar para que los formularios y las búsquedas funcionen tal y como exigían las especificaciones.



A la hora de cifrar por nosotros mismos la contraseña hemos utilizado Bcrypt que a diferencia de otros métodos de cifrados como SHA 256, este usa sal lo que dificulta los ataques con fuerza bruta.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Menú de navegación

El usuario normal y el administrador cuenta con menú de navegación de diferentes adecuado a sus necesidades:

El menú normal del usuario es este:

Texto

Descripción generada automáticamente

El menú de navegación del administrador es este:

Logotipo

Descripción generada automáticamente con confianza baja

La página también cuenta con un menú inferior que cuenta con los mismos campos que el menú superior.

#### Zona de empleado

El empleado tiene acceso a diferentes páginas con los recursos y formularios que el empleado necesitará:

##### Activación de la tarjeta bancaria

Los empleados tendrán acceso a la página donde podrán activar la tarjeta europea o internacional. Esto funciona gracias a un servicio de terceros que nos da servicios.

Hacemos una solicitud al API externo y nos devuelve los datos cifrados, la aplicación lo descifra con la contraseña dada y los muestra en pantalla con un formato parecido a una tarjeta de crédito real.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

##### Insertar gasto

Esta sección se compone de un formulario principal y 2 posibles formularios más según los datos que hayas insertado.

En el formulario principal se encuentran los campos de a qué departamento o proyecto se le imputa el gasto, fecha inicio, fecha fin y si se usara el transporte propio o público.

Si un empleado va a estar más de 4 días, se le aplicarán las dietas preestablecidas 60€ dentro de Europa y 100€ fuera de Europa y si el empleado va a estar fuera menos de 4 días la aplicación calculará los gastos en torno a los gastos introducidos en los formularios.

| Formulario de transporte público | Formulario de transporte propio |
| --- | --- |
|  |  |
| Si el transporte seleccionado ha sido público saldrán los campos relacionados con el transporte público. | Si el transporte seleccionado ha sido propio saldrán los campos relacionados con el transporte propio. |
| Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico  Descripción generada automáticamente | Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente |

Además, para ver si un país está en Europa o no para calcular las dietas. Hemos usado un API que nos brinda la información sobre los países. La API que hemos usado se llama Restcountries. Gracias a esta API podemos listar todos los países y así mostrar calcular si un país está dentro o fuera de Europa de forma automática.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

##### Fichar horas

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El empleado tendrá la opción de poder fichar horas. Las horas fichadas vendrán referenciadas con el proyecto en donde se han trabajado y el día en donde se ha trabajado.

Si un empleado trabaja más de 40 horas en una semana saldrá una advertencia. Aunque salte la advertencia el empleado podrá seguir fichando.

#### Zona de administrador

Las funciones del administrador son diferentes al de un usuario, por lo tanto, las páginas a las que tendrá acceso serán diferentes. Todas las páginas de administrador están restringidas y si un usuario sin privilegios intenta acceder a la ruta será redirigido de vuelta al perfil.

##### Gestión de usuarios

El administrador tiene acceso a la zona de gestión de usuarios. En esta zona el administrador podrá visualizar todos los usuarios. también tiene la opción de registrar un nuevo empleado, borrar uno existente, desactivar un usuario (por lo tanto no podrá hacer login) y modificar cualquier campo de administrador incluso la contraseña.Tabla

Descripción generada automáticamente

Una vez pulsamos el botón de crear y modificar la vista sería esta:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

##### Gestión de proyectos y departamentos

En esta página el administrador puede gestionar los proyectos y departamentos.

La primera tabla muestra los proyectos que existen y sus propiedades.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Aparte de ver los departamentos puedes modificarlos y cerrarlos por ejemplo (poniéndole una fecha fin). también se pueden eliminar y crear proyectos nuevos.

Más abajo en la página también te da la opción de gestionar los departamentos con otra tabla parecido a la anterior visual y funcionalmente:

Imagen que contiene Tabla

Descripción generada automáticamente

##### Integrantes de proyectosInterfaz de usuario gráfica, Aplicación Descripción generada automáticamente

En esta sección se podrán asociar empleados a proyectos existentes mediante un formulario con listas desplegables.

##### Buscador de gastos

En esta página el administrador podrá buscar los gastos solicitados. Se mostrarán todos los gastos, pero también existe la opción de filtrar, tanto por rango de fecha como por el costo del gasto.

#### Plugin

Para añadir funcionalidades a nuestra página de WordPress hemos usado varios plugin:

PHP everywhere: Este plugin sirve para añadir bloques código PHP a las páginas.

Ultimate member: Estamos usando este plugin para facilitar redirecciones con sesiones.

Limit Login Attemps Reloaded: Para limitar el número de intentos en el login de administrador

## Aplicación Android

Hemos desarrollado una aplicación Android desarrollada con Android Studio. Esta aplicación sirve para que los empleados puedan hacer diferentes tareas como activar la tarjeta de crédito, fichar horas, insertar gastos.

### Base de datos



Como sistema gestor de base de datos para la aplicación de Android. SQLite funciona como base de datos en local.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

### Navegación

#### Página de login

Esta es nuestra página de login puedes hacer login.

Aquí el usuario puede hacer login o registrarse. *La autenticación de usuarios funciona con el algoritmo de cifrado MD5.*

Gráfico, Gráfico de embudo

Descripción generada automáticamente

#### Menú principal

Este es el menú principal de la aplicación.

Desde aquí podrá ir a los siguientes apartados:

* Fichar

En esta pestaña el usuario puede introducir las horas trabajadas por cada empleado.

* Gestión de usuario

Aquí se podrán ver y modificar los usuarios que se encuentran en la base de datos.

* Perfil

Aquí se podrá ver la información de perfil del usuario.

* Tarjetas

Aquí se podrá cambiar la tarjeta de crédito de europea a internacional.

* Gestión de gastos

Aquí un usuario puede insertar los gastos que ha realizado.

* Proyectos

En esta pestaña el usuario puede ver crear y cerrar los proyectos. Además, también puede asignar los empleados a los diferentes proyectos.

* Acerca de

Damos información sobre la aplicación y la empresa.

* Salir

Es el botón de cerrar sesión.



#### Fichar

En esta sección el empleado puede fichar las horas trabajadas por el empleado.

Imagen que contiene Tabla

Descripción generada automáticamente

#### Perfil

En esta sección el usuario puede ver la información del propio usuario como, por ejemplo, nombre, apellido, nombre de usuario etc.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Tarjetas

En esta sección el usuario podrá cambiar entre las tarjetas europeas o internacionales.

Esto funciona consumiendo el API REST bancaria. Después de recibir la petición la aplicación desencripta la información y la muestra en pantalla.

##### Librería Volley

Para poder consumir hemos usado la librería volley que que facilita y agiliza el uso de redes en apps para Android. Esto nos permite la conexión con la API bancaria y recoger el string en JSON.

##### Conexión a internet

Para poder conectarnos a la API de internet tenemos que hacer una serie de modificaciones en el Android Manifest:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

#### Gestión de gastos

Para el formulario de gastos tenemos 2 formularios, uno si esta fuera más o menos de 4 días.

| Si el empleado va a estar menos de 4 días va a salir este formulario: | Si el empleado va a estar más de 4, saldrá este formulario: |
| --- | --- |
|  |  |

#### ProyectosInterfaz de usuario gráfica Descripción generada automáticamente con confianza media

Este apartado sirve para administrar proyectos.

Este apartado sirve para crear los proyectos.

En este cuadro de texto puedes cerrar un proyecto introduciendo el nombre del proyecto.



Aquí puedes introducir un usuario a un proyecto poniendo el nombre de usuario y con el de los proyectos.

#### Acerca deTexto Descripción generada automáticamente con confianza media

En esta pestaña puedes leer información sobre nosotros.

También puedes acceder a nuestros perfiles en redes sociales.

# Seguridad

Una de las prioridades de este proyecto ha sido la seguridad. Por lo tanto, hemos usado unas herramientas para conseguir que nuestras aplicaciones sean lo más seguras posibles.

## Ataques de reconocimiento

Los ataques de reconocimiento es la fase de preparación donde el atacante obtiene toda la información necesaria de su objetivo y/o víctima antes de lanzar un ataque. Es muy importante evitar esto ya que puede identificar los servicios y versiones que está usando el atacante. Para evitar esto hemos configurado una serie de contramedidas.

### Firewall

El firewall ha sido configurado de forma que desde fuera solo pueda conectarse al puerto del servidor web y que nosotros nos podamos conectar mediante SSH y se podrá acceder a la base de datos desde nuestra red (es la subred que da el wifi Maristak.IKT).

Texto

Descripción generada automáticamente

También el protocolo ICMP ha sido desactivado por lo que nuestro servidor ya no responde a ping:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

### Firma del servidor

La firma del servidor ha sido desactivada. Esto implica que no enseñara de forma tan pública la versión del apache2 a la hora de que exista un mensaje de error.

Esto se consigue poniendo en off en el atributo ServerSignature y prod en ServerTokens en el archivo del apache (*etc/apache2/apache.conf*). El resultado de la configuración es este:

| Antes de configurar | Después de configurar |
| --- | --- |
| Texto, Carta  Descripción generada automáticamente | Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente |

### Versión de wordpress oculta

Al inspeccionar la página se puede ver la versión del WordPress que se está usando, esto podría ser peligroso ya que podemos ver las vulnerabilidades que puede tener dicha versión. Esto se puede evitar añadiendo una función en functions.php:Texto

Descripción generada automáticamente

## Configuración del apache y WordPress

Hemos configurado algunos parámetros para garantizar la seguridad de nuestro servidor web.

### Certificado SSL

El certificado SSL se utiliza para establecer una conexión cifrada entre el servidor y el usuario mediante cifrado asimétrico. Esto imposibilita que los datos introducidos como contraseñas, métodos de pago … puedan ser obtenidos por terceros que están esnifando el tráfico.Texto

Descripción generada automáticamente

### Deshabilitar httpTexto Descripción generada automáticamente

Como no interesa que el servidor escucho http ya que tenemos https, hemos comentado entero la parte de virtualhost: \*80. Así el servidor web ya no escuchara el puerto 80.

### Directorios abiertos al servidor webTexto Descripción generada automáticamente

### Restricción de acceso a recursos dentro del WordPress

WordPress es un CMS muy completo y viene con varios archivos sensibles como el archivo de configuración o el directorio de recursos. Para esto hemos habilitado y configurado el archivo .htaccess.

Texto

Descripción generada automáticamente

Hemos denegado el acceso al archivo wp-config.php.

Con esta configuración nadie podrá acceder al archivo wp-config.php desde el navegador.

Hemos habilitado el acceso solo para los recursos que no interesan de wp-content.

Desde esta forma solo se podrá acceder a los recursos deseados al navegador y no se podrá acceder por ejemplo a functions.php que se encuentra dentro de este directorio, pero si podrá acceder a las css del tema.Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Cualquier intento a acceder los archivos protegidos por .htaccess resultaran en un error Forbidden 403:

### Eliminación de recursos no necesarios

Algunos elementos que vienen por defecto en la instalación en la instalación de WordPress. Los recursos que hemos borrado han sido wp-cron.php, install.php license.txt y readme.html.

### Permisos chmod

Todos los directorios tienen permisos 755 y todos los archivos tienen permisos 644.

Esto se ha hecho automáticamente usando:

find /var/www/wordpress -type d -exec chmod 755 {} \;

find /var/www/wordpress -type f -exec chmod 644 {} \;

### Deshabilitar trace/track

Trace and track son métodos que se encuentran por defecto habilitados. Estos métodos soportan “Cross-site” scripting lo que puede exponer las vulnerabilidades de nuestra web a posibles atacantes.

Esto se puede quitar poniendo esto en el archivo .htaccess:

RewriteEngine on

RewriteCond %{REQUEST\_METHOD} ^(TRACE|TRACK)

RewriteRule .\* - [F]

### Sentencias preparadas

Todas las interacciones con la base de datos de parte de la aplicación web usan sentencias preparadas. A parte de mejorar el rendimiento del servidor también mejora la seguridad de la aplicación ya que, dificulta en gran medida las inyecciones SQL. Las consultas preparadas no permiten a un potencial atacante desviar la consulta pasando parámetros que modifican su intención original.

Si un potencial atacante intenta hacer ataques como " OR 1=1 OR `name` LIKE " la base de datos interpretara " OR 1=1 OR `name` LIKE " como una cadena en vez interpretarlo como una sentencia SQL. Texto

Descripción generada automáticamente

### Temas en desuso

Hemos elegido el tema Astra y hemos borrado los otros temas que venían por defecto en WordPress.

### Errores de login personalizados

Por defecto las páginas de login de WordPress, muestran demasiada información. A la hora de realizar un intento de login fallido muestra que, si el usuario o contraseña es incorrecta, por lo tanto, hemos creado un error personalizado de fallo de inicio de sesión que no muestra si el parámetro fallido es el usuario o la contraseña:Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para poder cambiar esto he cambiado este parámetro en wp-includes/user.php:

Este es el error que sale si el usuario introducido está mal:

Captura de pantalla de computadora

Descripción generada automáticamente con confianza media

Este es el error que sale si la contraseña está mal:

Pantalla de computadora con fondo negro

Descripción generada automáticamente con confianza media

Ambos resultados son lo mismo para no enseñar el parámetro fallido.

### Intentos máximosInterfaz de usuario gráfica, Aplicación Descripción generada automáticamente

El intento de login máximo en el panel de administración han sido limitados. Un administrador tiene 3 intentos para acceder al login si falla la página de login se bloquea durante un tiempo.

## Vulnerabilidades OWASP

Hemos usado OWASP ZAP (Open Web Application Security Project) como herramienta de auditoría automática para detectar posibles vulnerabilidades que puede tener nuestro sitio.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Que es OWASP

OWASP es una organización sin fines de lucro, organizada por voluntarios. Proporcionan recursos gratuitos para la comunidad, como publicaciones, normas, software de testeo, software de capacitación…

### Top 10 vulnerabilidades OWASP 2021

Este es el top 10 vulnerabilidades OWASP 2021:

* A01:2021-Broken Access Control:Esta vulnerabilidad permite que un usuario pueda acceder a algún recurso o realizar alguna acción a la que se supone que no puede acceder.
* A02:2021-Cryptographic Failures: Esta vulnerabilidad significa que la criptografía no se implementó correctamente.
* A03:2021-Injection: Es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos.
* A04:2021-Insecure Design: Es una categoría amplia que representa diferentes debilidades, expresadas como "diseño de control faltante o ineficaz". Un diseño seguro aún puede tener defectos de implementación que conduzcan a vulnerabilidades que pueden explotarse. Un diseño inseguro no se puede arreglar con una implementación perfecta ya que, por definición, los controles de seguridad necesarios nunca se crearon para defenderse de ataques específicos.
* A05:2021-Security Misconfiguration: En esta categoría se contemplan los fallos que son causados por fallos de configuración como, por ejemplo, servicios no necesarios activados, cuentas por defecto activadas etc.
* A06:2021-Vulnerable and Outdated Components: Uso de software desactualizado, vulnerable o no compatible.
* A07:2021-Identification and Authentication Failures: Fallos relacionados con la autenticación, por ejemplo, permite los ataques de fuerza bruta, mal uso de recuperación de contraseñas, permite uso de contraseñas débiles etc.
* A08:2021-Software and Data Integrity Failures: Esta vulnerabilidad se relaciona con el código y la infraestructura que no protegen contra las violaciones de la integridad. Un ejemplo de esto es cuando una aplicación utiliza, bibliotecas o plugins que no son de confianza.
* A09:2021-Security Logging and Monitoring Failures: La falta de herramientas o mecanismos para registrar acciones relacionadas con la aplicación. Por ejemplo, inicios de sesión, inicios de sesión fallidos o transacciones de alto valor, no registradas.
* A10:2021-Server-Side Request Forgery: Si un atacante puede hacer que el servidor emita (normalmente) solicitudes HTTP en su nombre, entonces eso es SSRF. Esta vulnerabilidad. Permite realizar solicitudes de forma gratuita a cualquier cosa que el servidor pueda ver.

### Vulnerabilidades muy graves

Nuestra página contaba con un error muy grave de “path traversal” identificado con CWE ID 22 y WASC 33, este problema ha sido solucionado borrando el directorio wp-json que no estábamos usando para nada.

Esta vulnerabilidad permite a los atacantes realizar un ataque de path traversal. Que permite el acceso a archivos y directorios que están almacenadas fuera del document root de nuestro sitio.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

### Vulnerabilidades graves

#### X-Frame-Options Header Not Set

Tenemos algunos fallos relacionados con el atributo X-Frame que no está configurado. Los identificadores de este error son, CWE ID 1024 y WASC ID 15. Esto permite que se use los renderizar una página en las etiquetas <frame>, <iframe> o <object> lo que puede conllevar ataques de clickjacking.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Esto puede ser un ataque en el que se engaña a un usuario para que haga clic en contenido procesable en un sitio web oculto haciendo clic en algún otro contenido en un sitio web señuelo.

Este error ha sido arreglado con una función en functions.php:

*function add\_header\_xframeoptions() {*

*header('X-Frame-Options: SAMEORIGIN');*

*}*

*add\_action('send\_headers', 'add\_header\_xframeoptions’);*

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Configurar este mismo parámetro en más de un sitio también da una advertencia de seguridad, pero esto se soluciona poniéndolo en solo un sitio.

#### Application error disclosure

Esta vulnerabilidad está identificada con CWE ID 200 y WASC ID 13. Este fallo era que al apuntar a los directorios enseñaba el listado de los archivos que se encontraban en el directorio, esto implicaba que los usuarios pueden ver los listados de los archivos del directorio wp-includes, ya que no estaba correctamente configurado.

Esto se soluciona quitando la opción de listar los archivos en la configuración del directorio.

Texto

Descripción generada automáticamente

Poniendo Options -Indexes deshabilitamos la opción de listar los ficheros.

## Vulnerabilidades API REST

La API REST bancario con el que trabaja nuestras aplicaciones cuenta con varias fallas de seguridad muy importantes:

Pese a que la aplicación cuenta con una contraseña para hacer los request, la API no cuenta con autenticación. No cuenta con cuentas de usuario con contraseñas para que pueda limitar el acceso a personas malintencionadas. Un usuario que sepa la ruta de la API podría hacer solicitudes solamente apuntando al link.

Otro fallo de seguridad que cuenta la API es que las comunicaciones no están cifradas, funciona con http. Las comunicaciones entre nuestro sitio y la API pueden ser esnifadas. La propia API ya cuenta con cifrado AES 256 por lo que un programa que esnifa tráfico no podría conseguir la información en texto claro pero el cifrado https le daría una capa más de cifrado. Aparte, la propia contraseña no la consideraríamos lo suficientemente segura para solo confiar en ella en una API relacionada con tarjetas de crédito.

# Conclusiones

Aunque no hemos podido cumplir todos los requisitos estamos contentos con el resultado de la aplicación multiplataforma. La aplicación web es totalmente funcional y la aplicación Android es la única característica que impide que la aplicación no sea multiplataforma es que la base de datos de MariaDB y SQLite no están sincronizados.

Para muchos integrantes del grupo, este ha sido el proyecto más grande de desarrollo y creemos que en nuestro grupo nos hemos integrado bien.

Por todo ello creemos que hemos cumplido con los objetivos y estamos muy satisfechos con el resultado.

## Líneas futuras

A lo largo de estos meses de proyecto hemos tenido varios problemas y objetivos que no hemos podido cumplir.

* No hemos podido sincronizar la base de datos de MariaDB con la base de datos Android SQLite.
* Tuvimos un gran problema con las sesiones de wordpress. Lo que nos hizo perder mucho tiempo.
* La aplicación móvil no presenta distinción de roles
* No hemos configurado una pestaña de ajustes en la aplicación Android.
* No pudimos configurar la subida de tickets a la hora de ingresar gastos en la aplicación web.
* En la aplicación Android no hemos conseguido que el usuario pueda poner una foto de perfil.

# Referencias

<https://developer.android.com/>

<https://www.hostinger.es/tutoriales/que-es-apache/>

<https://geekflare.com/es/apache-web-server-hardening-security/>

<https://www.zaproxy.org/docs/guides/zapping-the-top-10-2021/>

<https://owasp.org/www-pdf-archive/Introduccion_a_la_OWASP.pdf>

<https://owasp.org/Top10/>

<https://stackoverflow.com/>

<https://www.youtube.com/channel/UCeEW6PHNJlIwI-rKChmqqkw>

<https://www.youtube.com/user/CristianDavidHenao>

<https://www.youtube.com/channel/UCsOnfOiyP-ftrVTyORYxurA>

# Anexos

Link al repositorio de GitHub:

<https://github.com/aimenrial432/reto_ciber1_equipo5>