

SECURE SOFTWARE DEVELOPMENT



Índice

Secure software development

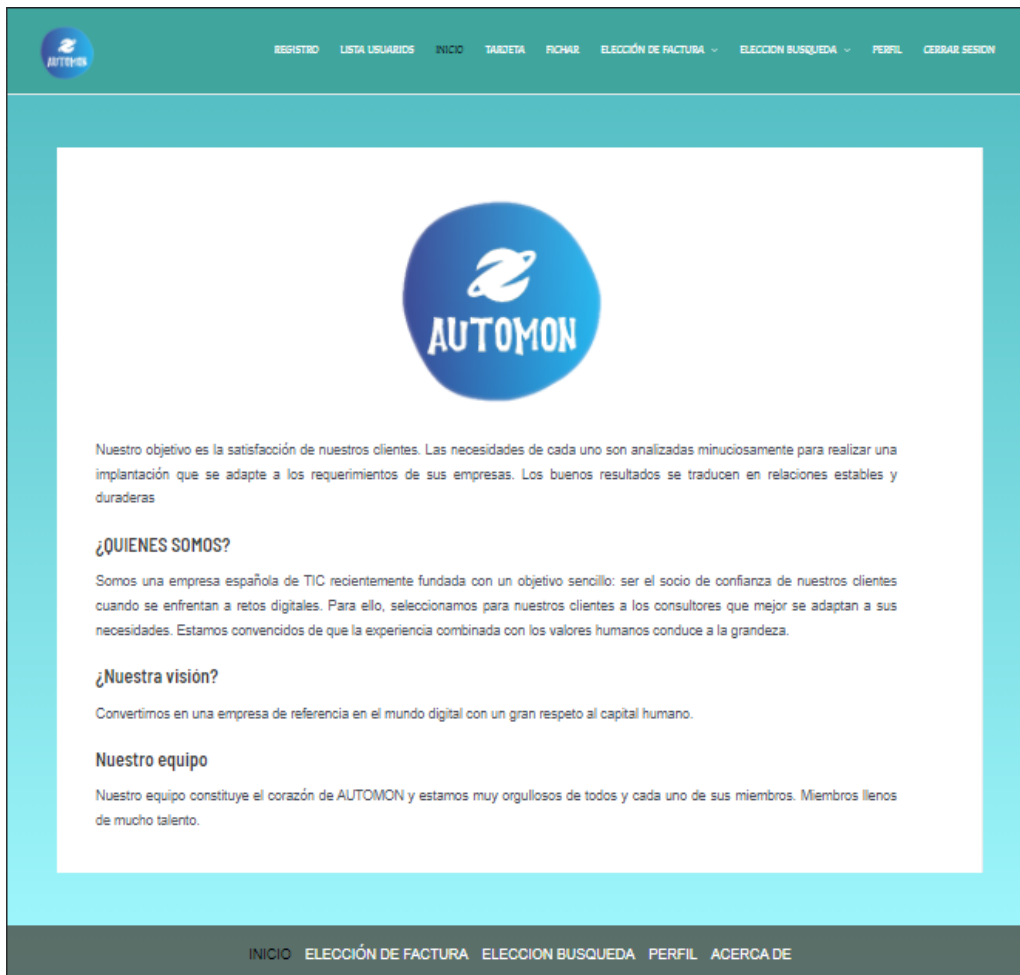
1.Introducción.....	3
2.Objetivos.....	4
2.1_Arquitectura.....	4
2.2_Especificaciones.....	5
3.Sitio web	8
3.1 Características técnicas	8
3.1.1_Sistema operativo.....	8
3.1.2_Gestor de contenidos	8
3.1.3_Gestor de base de datos Mariadb.....	8
3.1.4_Navegación del sitio web	11
3.1.5_Validaciones.....	12
3.1.6_Sitio web adaptable a tamaños diferentes.....	13
3.1.7_Autenticación.....	13
3.1.8_Consumo de Web Service.....	14
3.1.9_Plugins utilizados	15
3.2 Funcionamiento y paginas	16
4.Aplicacion Android	24
5. Programas o plataformas utilizadas	25
6.Securizacion	26
6.1 Securizacion del servidor.....	26
6.1.1_Credenciales de cuentas.....	26
6.1.2_Bloqueo puertos, Firewall	27
6.1.3_Protocolo seguro y certificación (SSL y Https).....	28
6.1.3_Securizacion gestor de base de datos	30
6.1.4_Securizacion gestor de contenidos (WordPress)	32
6.1.5_Securizacion ficheros y directorios	35
6.2 Validación	39
6.3 Mejoras API REST banco.....	42
7.Informacion entregada.....	44
8.Conclusion.....	45
9.Referencias	46

1.Introducción

En la **documentación** realizada del **proyecto** por los alumnos Liher Ramoneda, Andoni Uribe y Jon Herrero para el reto presentado por Maristak Ikastetxea Durango.

Se trata de un **proyecto** de carácter **académico** en el cual el objetivo es crear un sitio web y una aplicación Android para la gestión de gastos de una empresa que se puso en contacto con nosotros.

En este **documento** se recogen las **especificaciones** técnicas y principales características del proyecto.



2.Objetivos

El **objetivo** principal del proyecto es el **desarrollo** de un sitio web y una aplicación Android para la gestión de gastos sobre viajes de la empresa que satisfaga las especificaciones funcionales y no funcionales exigidas por la empresa que lo solicita.

La **aplicación** debe **gestionar** los gastos y dietas derivados de los viajes de los empleados y la activación de las tarjetas de crédito corporativas, europea o internacional.

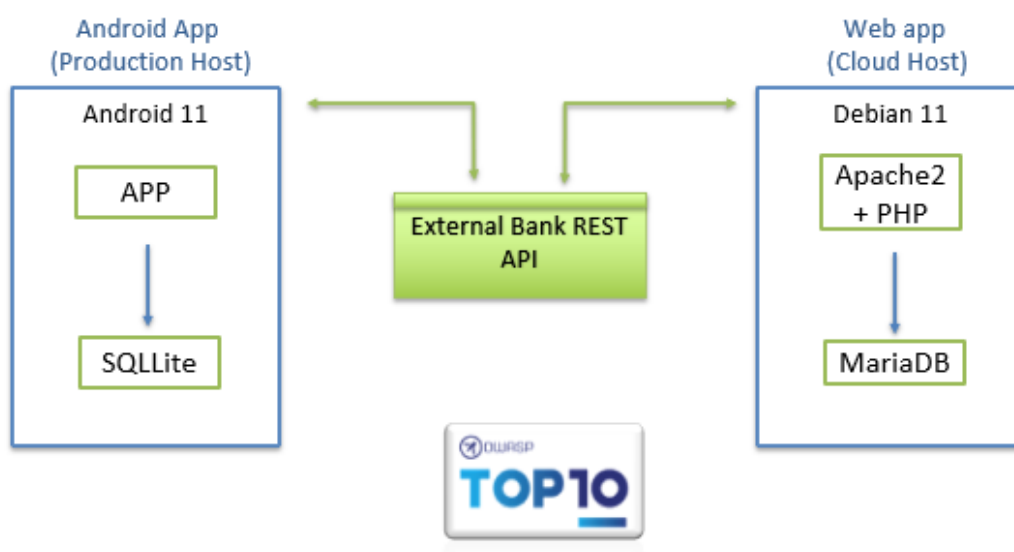
2.1_Arquitectura

La **arquitectura** principal que debe contener el proyecto es el siguiente:

Principalmente existen dos hosts un servidor que es Debian en el hipervisor de VMware esxi el que contiene los servicios apache2, PHP, wordpress, MariaDb el cual está en la nube.

Por el otro lado el host de Android en el cual ira la aplicación Android y una base de datos local con SQLite. Además de tener otro servicio de banco API REST el cual enlazaremos con los dos hosts para que puedan acceder al contenido del banco, para la activación de las tarjetas.

Principalmente teníamos pensado en sincronizar Mariadb con la base de datos SQLite, pero por problemas en la conexión de SQLite al sincronizar con Mariadb, preferimos dejarlo de manera local en el host servidor y en el host Android.



2.2_Especificaciones

Especificaciones generales Web/APP

- **DESARROLLO:** aplicación Android y sitio web.
- **NAVEGACIÓN SUPERIOR:** menú, lista ...
- **NAVEGACION INFERIOR:** enlaces a las diferentes páginas de la aplicación (Inicio, Gastos, Perfil ...).
- **TOOLBAR:** configuración / “acerca de”.
- **ROLES:** roles específicos (1-empleado, 2-administrador)
 - Empleados: podrán registrar los datos asociados a su actividad.
 - Administrador: podrán realizar consultas, además de gestionar altas, bajas y modificaciones.
- **Sección perfil de usuario.**
 - Nombre y apellidos
 - DNI
 - Nombre de usuario
 - Última conexión
- **Secciones de las páginas:**
 - Gestión de gastos.
 - Gestión de dietas.
 - Gestión de tarjetas (Web services).
 - Control de fichaje de empleados.
 - Búsqueda de gastos (Por fecha e importe).
 - Inicio de sesión de usuario.
 - Perfil.
 - Cerrar Sesión.
 - Búsqueda de gastos.

Especificaciones gestión de gastos y proyectos

- Se tienen que poder introducir los gastos producidos para aprobación. Nos han indicado que hay que adjuntar los tickets. Es requisito almacenar estos datos:
 - Fecha y hora en que se han producido
 - Medio de transporte
 - Distancias recorridas
 - Peaje
 - Parking
 - Dieta
 - Otros conceptos
- Una vez introducidos estos datos deberá calcular la suma total de los gastos generados. El precio que se paga por kilometraje es de 0.3€/km.
- Si un empleado va a estar fuera más de cuatro días directamente puede solicitar dietas sin tener que introducir los gastos. Estas son las especificaciones:
 - Tiene que indicar fecha inicio y fecha fin.
 - Tiene que indicar en qué país y ciudad se encuentra.
 - Las dietas definidas por la empresa son:
 - 60€ día en Europa
 - 100€ día fuera de Europa
 - La aplicación deberá de calcular el importe total de los gastos.
- Tiene que contener una sección para incluir las horas que se han metido en cada proyecto cada día. Esto significa:
 - La jornada laboral es de 40 horas semanales.
 - Se debe de poder registrar las horas trabajadas por día.
 - Se debe de poder registrar las horas asignadas a cada proyecto.
- Ambas opciones deberán de incluir lo siguiente: el proyecto al que se ha imputado el gasto. Actualmente hay diferentes cuentas a las que imputar:
 - DEP: Departamento. Indicar a través de una caja de texto el código del departamento.
 - PRO: Código de proyecto. Indicar a través de una caja de texto el código del proyecto.
- Deberá de incluir una opción de consulta de datos, donde el usuario pueda consultar el histórico de datos que ha introducido desde la aplicación. Esta pantalla, deberá incluir los siguientes filtros de búsqueda:
 - Rango de fechas: Fecha Inicio y Fecha Fin.
 - Rango de importes totales (suma de parking, más dietas,...). P.e→ Entre 40€ y 60€

Especificaciones técnicas

Sitio web / Android:

- Plataforma Android:
 - Versión: 8.0 hacia adelante.
- Base de datos local:
 - Android: SQLite.
 - Versión: 3.37.0.
 - Sitio web: Mariadb.
 - Versión: Ver 15.1 Distrib 10.3.31-MariaDB
- Servidor web: Apache2.
 - Versión: Apache/2.4.38 (Debian).
- Integracion Web Servicios del Banco Api Rest.
- CMS (Gestor de contenidos): WordPress.
 - Versión: current 5.8.2
- OS (Sistema Operativo) servidor: Debian.
 - Versión: 10
- Aplicación Android estudio.
 - Versión: 2020.3.1
- GitHub
 - Versión 2.9.5
 - Repositorio: Android, PHP
- Java
 - Versión: 11.0.10
- PHP
 - 7.3

3.Sitio web

3.1 Características técnicas

3.1.1_Sistema operativo

Hemos decidido escoger Debian 10 para alojar el sitio web, por las siguientes razones.

Debian es una distribución más estable y pura que Ubuntu. Y esto se traduce en un **mejor rendimiento y una mayor estabilidad**, además de un **soporte** mucho más extendido.

El sistema Debian 10 ira alojado en una máquina virtual esxi ubicada en la dirección 10.122.24.251

El host del servidor Debian tendrá la siguiente dirección Ip: 10.122.27.103

3.1.2_Gestor de contenidos

Hemos utilizado el gestor de contenidos de WordPress para el desarrollo del sitio web.

Esta elección está basada en las siguientes características:

- ◇ Gratuito y de código abierto.
- ◇ CMS más utilizado en el mundo.
- ◇ Sencillez.
- ◇ Personalización completa mediante temas y plugins.
- ◇ Seguridad y mantenimiento.

Finalmente los hemos escogido este porque es compatible con el gestor Mariadb y con los servicios apache del sistema operativo Debian.

3.1.3_Gestor de base de datos Mariadb

Elegimos Mariadb como base de datos por los siguientes motivos:

WordPress, funciona de forma nativa, puede trabajar con **bases de datos MySQL** y también con **Mariadb**, ya que es un **fork de MySQL** y son totalmente compatibles.

Además, mediante librerías externas también **podemos hacer que WordPress funcione con SQLite**, muy útil para sincronizar la base de datos.

Los administradores se conectarán mediante **MySQL Workbench** para la creación de las tablas y campos de la base de datos, o para el mantenimiento de la misma.

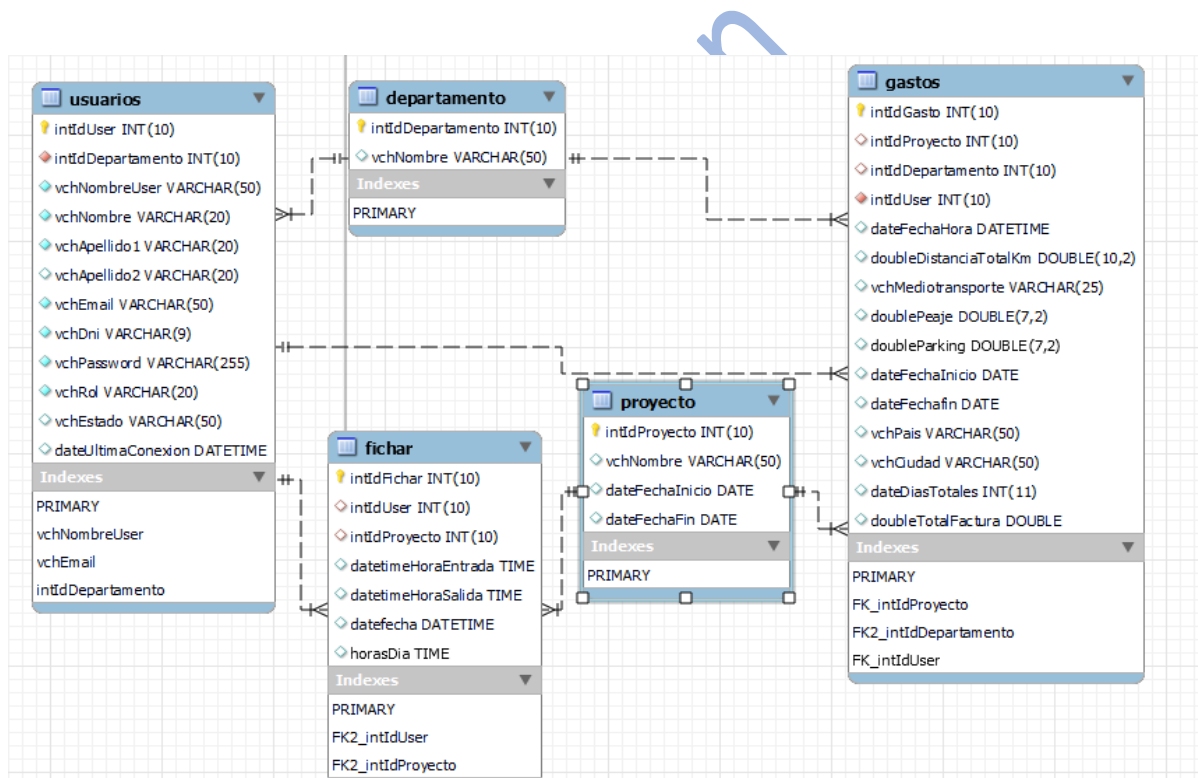
Creación esquema base de datos

A la hora de crear la base de datos para la **gestión de gastos**, decidimos hacerlo de la siguiente manera.

En primer lugar, el diseño de base de datos será parecido para el gestor SQLite y Mariadb, ya que van a utilizar las mismas tabla y campos, estas estarán de forma local, tendremos una bdd local para el sitio web y otra para la aplicación Android

En el diseño las **tablas** y los **campos**, además de las relaciones con las otras tablas. Principalmente teníamos decidió crear 5 tablas.

- La primera tabla usuarios
- La segunda tabla departamentos
- La tercera tabla proyectos
- La cuarta tabla fichajes
- La quinta tabla gastos



Estas **tablas** tienen los siguientes **datos** que almacenaran al **crear** o **añadir** gastos desde la aplicación o sitio web:

- **Tabla de usuarios:** aquí se introducen los datos del empleado y también los usuarios, esta tabla tiene un identificador "intIdUser", el cual nos permite saber que usuario realiza la diferentes acciones gastos, dietas, fichajes, etc... y en el cual se nos muestra información acerca del empleado.
- **Tabla de departamento:** aquí se introducen los datos asociados a los departamentos de la empresa. En esta tabla se almacenan dos datos, el identificador "intIdDepartamento" y el nombre.
- **Tabla de proyectos:** aquí se nos muestran los datos asociados a los proyectos de la empresa, con fecha inicial y final.
- **Tabla fichar:** aquí se nos muestran y los usuarios añaden el tiempo de trabajo al proyecto en el que estén involucrados.
- **Tabla gastos:** aquí se añaden los datos que los usuarios añaden introduciendo los gastos mediante la aplicación y se almacenan. Por ejemplo: las dietas, gastos mas de 4 días, menor a 4 días, total de los gastos, etc..

3.1.4_Navegación del sitio web

En cuanto a los componentes de navegación, hemos optado por simplificar los menús haciéndolos de forma sencilla y fáciles de identificar, además, en las especificaciones del cliente nos piden una barra superior y una barra inferior.

Hemos utilizado la barra superior para colocar en ella dos menús desplegables y la barra inferior para el acceso rápido a las opciones, las cuales podríamos decir que son el núcleo de la aplicación, como son el registro de gastos y el registro de dietas.

Al interactuar con cada página del menú, el usuario accederá a estas, en las cuales estarán las diferentes funciones que requiere el cliente, por ejemplo:

- Añadir los gastos.
- Fichaje del empleado.
- Perfil.
- Búsqueda de gastos.
- Etc...

Visual y funcionalmente estarán creadas de la siguiente manera:

- Navegación superior, con 2 menús:



- Navegación inferior:



En la siguiente imagen vemos la pantalla correspondiente a la búsqueda de facturas con las barras de navegación superior e inferior implementadas.

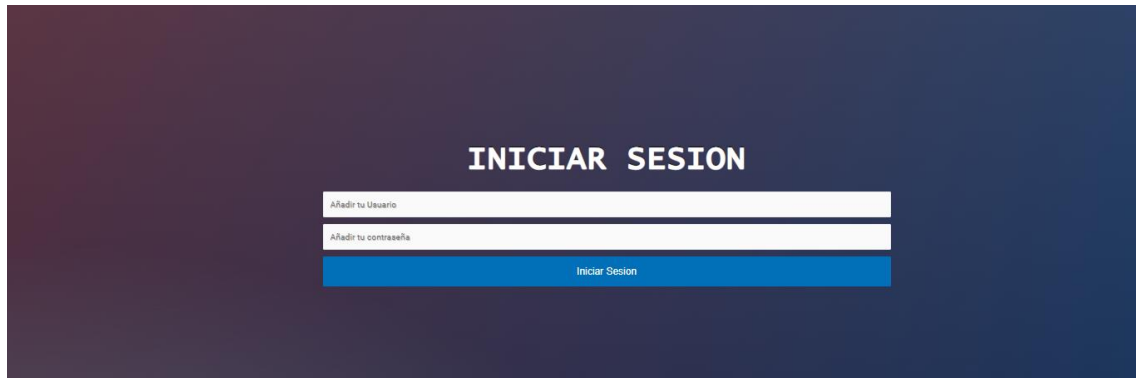


3.1.5_Validaciones

El sitio Web consta de validaciones para asegurar la funcionalidad asegurándose de que los datos insertados son corrector y válidos.

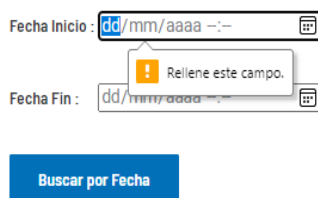
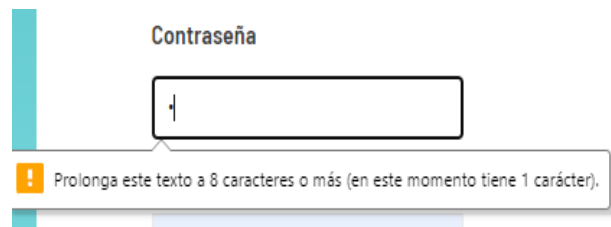
Procesos en el acceso a la página:

- Solo puedan acceder y visualizar la página los usuarios que inicien sesión
- Los usuarios empleados no tienen acceso a utilizar las páginas enfocadas al administrador (Listar usuarios, modificar usuarios y desactivar usuarios)



Procesos validación formularios:

- «Este campo es obligatorio» (No se puede dejar este campo en blanco).
- «Introduzca el DNI en el formato 79112324X» (Se requiere un formato de datos específico para que se considere válido).
- «Introduzca un horario el cual no supere las 8 horas de trabajo».
- «Su contraseña debe tener entre 8 y 30 caracteres y contener una letra mayúscula, un símbolo y un número». (Se requiere un formato de datos muy específico para tus datos).

3.1.6_ Sitio web adaptable a tamaños diferentes

El sitio web está desarrollado de modo que sea responsive, esto significa que se verá correctamente en diferentes dispositivos.

Fecha del gasto	Proyecto	Departamento	Días de Viaje	Distancia recorrida	Medio de Transporte	Precio Peaje	Precio Parking	Factura total
2021-11-16 13:08:24	Omega	Finanzas		100.00	taxi	100.00	100.00	230
2021-11-16 13:08:35		Desarrollo		10.00	tren	10.00	11.00	24
2021-11-16 13:08:50	Aurora			170.00	coche	12.00	11.00	74
2021-11-16 13:09:08	Omega		29					2900
2021-11-16 13:09:20		Desarrollo	1					100

2021-11-16 13:08:24
Omega
Finanzas
100.00
taxi
100.00
100.00
230

3.1.7_Autenticación

Mediante un inicio de sesión comprobamos el usuario y la contraseña insertados por el usuario coinciden con los datos de la tabla usuario en la base de datos. En caso de error, el login nos mostrará un mensaje el cual no permite acceder a la página web, en caso contrario, podrá acceder a la web.

Por otra parte, el usuario puede estar desactivado, en ese caso tampoco dejará acceder al sitio web.

Por último, cuando el usuario haya iniciado sesión correctamente, se guardará en la base de datos la fecha y hora del login, que será la de la última conexión la próxima vez que haga login.

INICIAR SESION

Iniciar Sesion

3.1.8_Consumo de Web Service

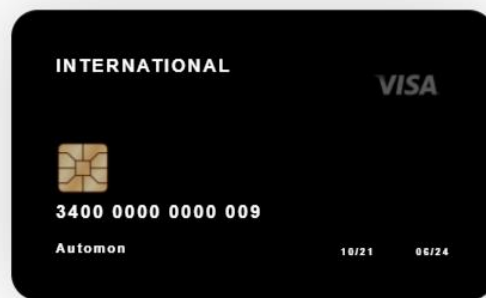
El sitio web al igual que la aplicación móvil simula el consumo de un Web Service que ofrece el banco ficticio para la activación de tarjetas europea e internacional. Para ello utilizaremos código PHP con el que consumiremos el contenido API de la tarjeta.

Principalmente el código PHP recoge los datos de la URL definida, y desciframos el código de base64. Con la clave facilitada, desciframos y se nos mostrará en AES(Base64) 256 bits y para después traducirlo a texto plano y entonces tendremos la información que nos permitirá recogerla con variables.

Una vez hecho esto, mostramos en una tarjeta creada con CSS en la cual, con variables, añadimos los datos recogidos de la API.

ACTIVAR TARJETA

Europe



3.1.9_Plugins utilizados

En esta sección se muestran los plugin utilizados en la página web.

Plugin seguridad:

- ◇ **Akismet Anti-Spam** incluye un conjunto de herramientas para la lucha efectiva contra el spam de los comentarios y trackbacks.
- ◇ **Limit Login Attempts Reloaded** se encarga de limitar los intentos de inicio de sesión proteger el sitio de posibles ataques de fuerza bruta.
- ◇ **WP Advanced Math Captcha** se encarga de limitar el acceso a los intentos de inicio de sesión de bots, añadiendo medidas de sumas y restas.

Plugin copias de seguridad y BackUp:

- ◇ **UpdraftPlus permite** realizar copias de seguridad de todos los archivos de tu web y también de su base de datos, guardarlas en la nube y restáuralas a golpe de clic.

Este plugin se conecta con numerosos servicios de almacenamiento online en las que guardar tus copias, tales como Dropbox, Google Drive, por correo electrónico...

Otros plugin:

- ◇ **Child Theme configurator** para crear y configurar temas hijos o "child themes" en WordPress.
- ◇ **SiteOrigin CSS** editor de css para maquetar la forma visual de las diferentes páginas de wordpress.
- ◇ **PHP Everywhere** editor el cual nos permite insertar código php y html en páginas, widgets o posts en wordpress.
- ◇ **Ultimate member**, permite crear formularios y redirecciones a otras páginas.

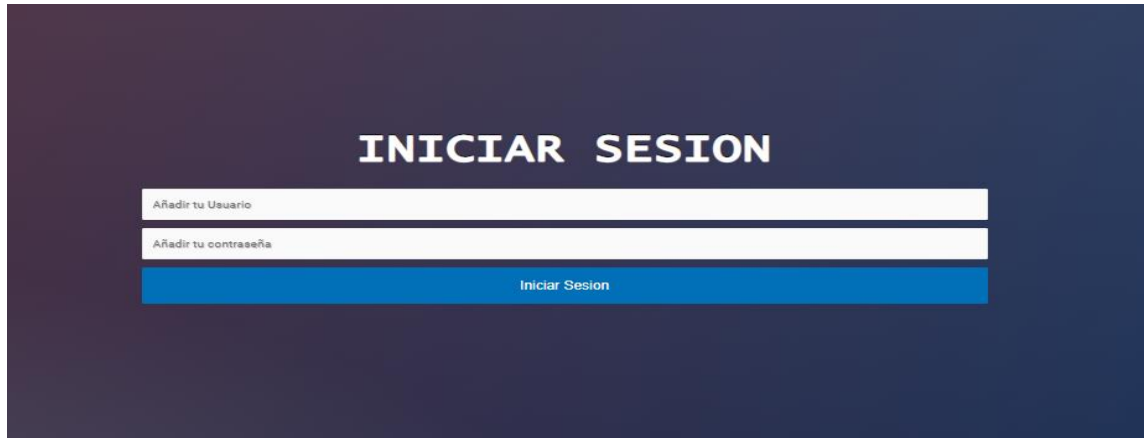
3.2 Funcionamiento y paginas

Login

La primera página que se muestra al acceder desde la URL será la de login, cuando accedamos al sitio web nos pedirá iniciar sesión, para poder acceder a las funcionalidades.

En la página login tiene la siguiente forma.

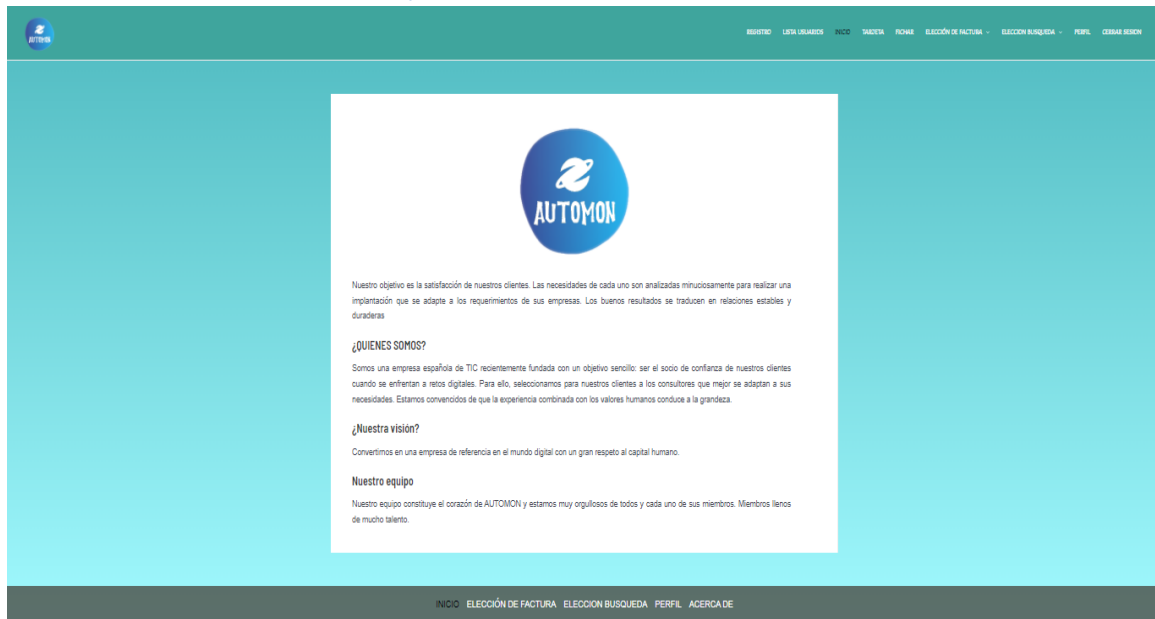
Solo se mostrará un formulario con usuario y contraseña.



Inicio

Al iniciar sesión dirigirá al usuario a la página principal inicio, en la que se detalla información acerca de la empresa y de la aplicación.

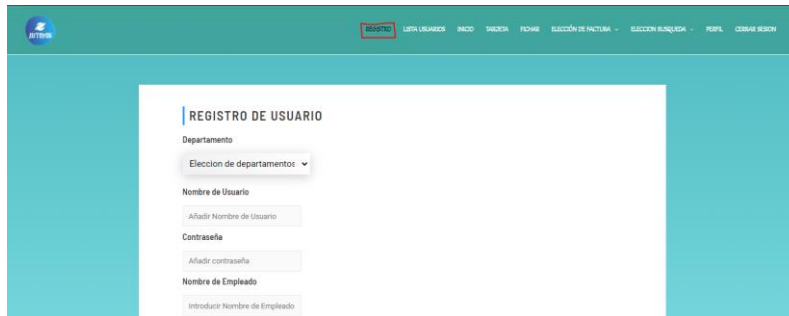
Además de poder acceder al contenido de la aplicación, gastos, búsqueda, tarjeta...



Registro Usuarios

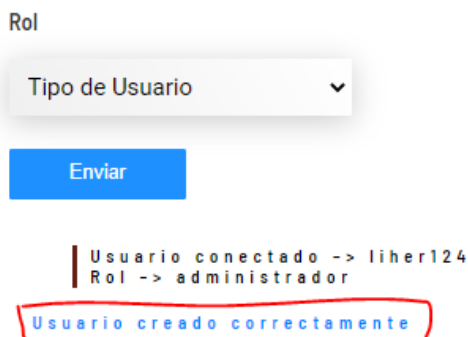
Para registrar usuarios accedemos a la barra de navegación superior o inferior y accederemos a la página “Registro”.

Únicamente los usuarios con permisos de administrador podrán acceder a este apartado.



Una vez en el formulario, rellenamos los datos de los usuarios y le damos al botón enviar.

La aplicación nos mostrará un mensaje de que se ha creado el usuario.



REGISTRO DE USUARIO

Departamento

Eleccion de departamentc ▼

Nombre de Usuario

Añadir Nombre de Usuario

Contraseña

Añadir contraseña

Nombre de Empleado

Introducir Nombre de Empleado

Primer Apellido

Introducir el Primer Apellido

Segundo Apellido

Introducir el Segundo Apellido

DNI

Añadir el DNI

Email

Añadir el correo

Rol

Tipo de Usuario ▼

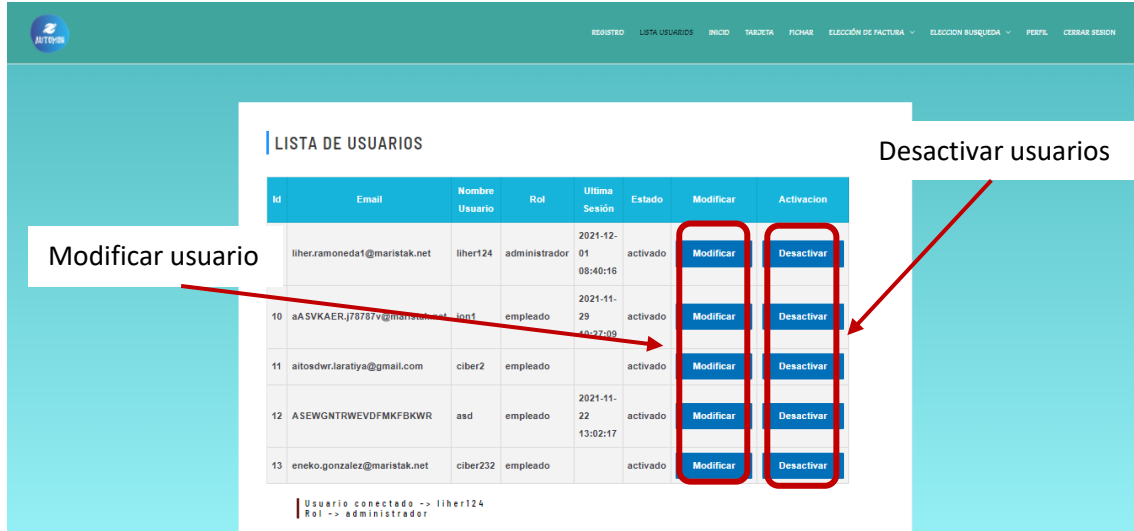
Enviar

Usuario conectado -> liher124
Rol -> administrador

Lista de Usuarios

Para obtener un listado de los usuarios, accedemos desde el menú desplegable en la barra de navegación superior.

En esta página solo podrá acceder con el rol del administrador, aquí se listan todos los usuarios de la app y web. Además de poder desactivar usuarios en el caso que dejen de baja y modificar sus respectivos datos.



Id	Email	Nombre Usuario	Rol	Última Sesión	Estado	Modificar	Activación
	liher.ramonedat1@maristak.net	liher124	administrador	2021-12-01 08:40:16	activado	Modificar	Desactivar
10	aASVKAERj78787v@gmail.com	jon1	empleado	2021-11-29 10:57:09	activado	Modificar	Desactivar
11	altosdwr.jaratty@gmail.com	ciber2	empleado		activado	Modificar	Desactivar
12	ASEWGNTRWEVDFMFKFBKWR	asd	empleado	2021-11-22 13:02:17	activado	Modificar	Desactivar
13	eneko.gonzalez@maristak.net	ciber232	empleado		activado	Modificar	Desactivar

Usuario conectado --> liher124
Rol --> administrador

Activación tarjeta

Para acceder a la gestión de las tarjetas de crédito accederemos a la barra de navegación superior o inferior y accederemos a la página "Tarjeta", en dicha página nos dará la opción para activar la tarjeta internacional o europea.

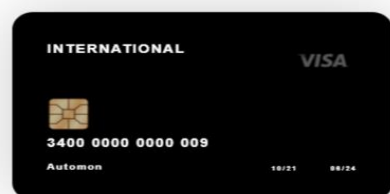
Se mostrará una ilustración de la tarjeta en la cual se mostrarán los datos obtenidos del Web Service de la API de la tarjeta:

- Número de la tarjeta.
- Fecha de caducidad.
- Tipo de tarjeta (internacional o europea).
- Nombre de la empresa.

Hay que tener en cuenta que la activación de una tarjeta supone la desactivación de la otra.

ACTIVAR TARJETA

Europe

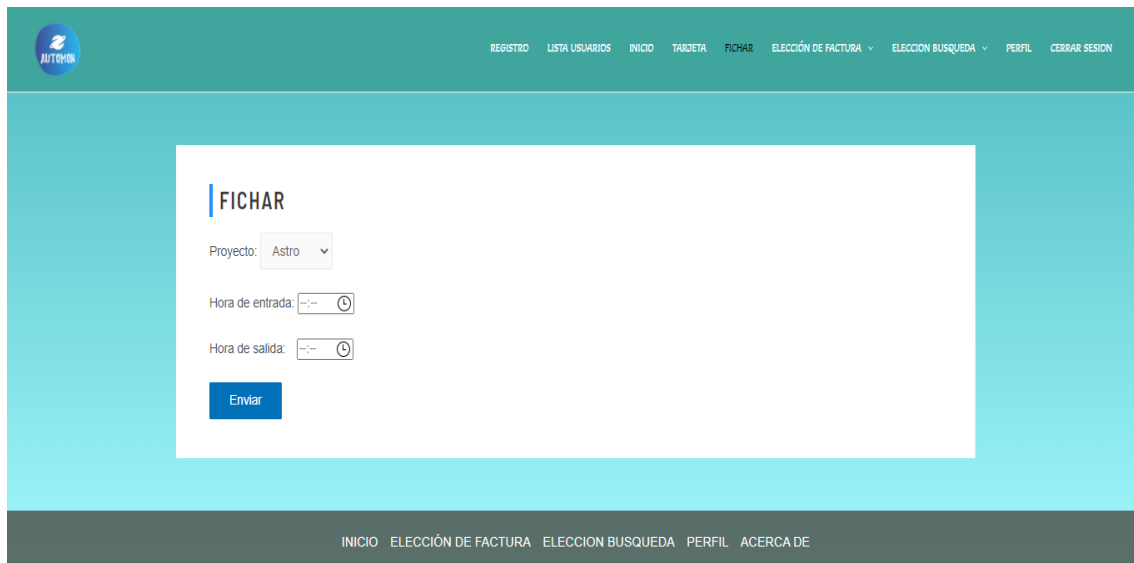


Fichar

Para utilizar la función de fichar de usuarios accederemos a la barra de navegación superior y nos introduciremos a la página “Fichar”.

Una vez en el formulario rellenamos los datos de la hora entrada y salida. Además, también introduciremos al proyecto que se le imputara las horas que trabajemos.

Una vez imputadas las horas la aplicación nos mostrará un mensaje de que se ha añadido correctamente al proyecto.



FICHAR

Proyecto: Astro ▼

Hora de entrada: --:-- ⌚

Hora de salida: --:-- ⌚

Enviar

INICIO ELECCIÓN DE FACTURA ELECCION BUSQUEDA PERFIL ACERCA DE

Gastos

Para registrar un gasto accedemos a la barra de navegación superior o inferior y accederemos a la página “Elección de Facturas”, en dicha página nos dará la opción para añadir gastos o dietas.

Para registrar un gasto nos dirigiremos a la barra superior, y haremos clic en la sección “Elección de facturas”. En dicha página, filtrar el tipo de gasto que realizaremos por los días de duración de dicho viaje, “menos de 4 días” nos dirigirá a “Gastos” y la opción de “más de 4 días” nos dirigirá a “Dietas”.

En este caso accederemos a la opción de “Estar menos de 4 días”.



Una vez en el formulario rellenamos los datos de los gastos y le damos al botón enviar.

La aplicación nos mostrará un mensaje de que se ha creado el gasto.

Parking:

 Factura realizada correctamente

GASTOS

Seleccione el departamento:

Eleccion de departamentos ▼

Seleccione el proyecto:

Eleccion de proyecto ▼

Distancia:

Añadir los kilometros recorridos

Seleccione el transporte utilizado:

Eleccion de Transporte ▼

Peaje:

Precio del Peaje

Parking:

Precio del Parking

Dieta

En esta ocasión, seleccionaremos la opción de “estar más de 4 días”, donde podremos insertar dietas.



Se añadirán los siguientes datos: departamento o proyecto, país, ciudad fecha inicial, fecha final y la tarjeta.

Una vez en el formulario rellenamos los datos de las dietas y le damos al botón enviar.

La aplicación nos mostrará un mensaje de que se ha creado las dietas.

dd/mm/aaaa

Eleccion de Tarjeta

Insertar Dieta

Dieta Internacional insertada correctamente

DIETA

Seleccione el departamento:

Eleccion de departamentos

Seleccione el proyecto:

Eleccion de proyecto

País:

Añadir el pais

Ciudad:

Añadir la ciudad

Fecha Inicio:

dd/mm/aaaa

Fecha Fin:

dd/mm/aaaa

Eleccion de Tarjeta

Insertar Dieta

Búsquedas de gastos

Para consultar las dietas o gastos que se han realizado accederemos a la barra de navegación superior o inferior y accederemos a la página “Elección de Búsqueda”, en dicha página nos dará la opción de buscar por Fecha o Importe.

Una vez en el formulario, nos presentará los campos donde introducir los criterios de filtro o búsqueda. Podemos acotar el filtro mediante 2 fechas, o en caso de importe, mediante 2 valores. También es posible combinar cualquiera de los cuatro parámetros, desde dejar vacíos o rellenar los 4.

Al momento de pulsar el botón de “Buscar” se ejecutará una consulta con los parámetros introducidos en el formulario y nos mostrará un listado con las filas que cumplan los criterios indicados. Si no hay ninguna coincidencia el listado se mostrará en blanco.

Búsqueda por fecha o importe:

BUSQUEDA FACTURAS POR FECHAS

Fecha Inicio:

Fecha Fin:

Buscar por Fecha

Usuario conectado -> liher124
Rol -> administrador

BUSQUEDA DE GASTOS POR IMPORTE

Importe Inicial:

Importe Final:

Buscar por Importe

Usuario conectado -> liher124
Rol -> administrador

Perfil

Para consultar los datos del usuario conectado o “Perfil” accedemos a la barra de navegación superior o inferior y accederemos a la página “Perfil”.

Esto una vez accedamos nos mostrará una pantalla con la foto y alias del usuario, nombre y apellido, DNI, fecha y hora de la última vez que se conectó o utilizó la aplicación.

Informacion del Perfil



Nombre Apellidos	Dni	Nombre Usuario	Última Conexion
Liher Ramoneda	79116378N	liher124	2021-11-24 09:39:24

Usuario conectado -> liher124
Rol -> administrador

Cerrar Sesión

Al pulsar en esta sección el usuario cerrará sesión y le redirigirá a la página de inicio de sesión.

Acerca de

En esta página se mostrará información de la versión del sitio web y otros datos acerca del sitio web.

Automon

4.Aplicacion Android

Automon

5. Programas o plataformas utilizadas

- IDE Android Studio
- IDE Eclipse (java, php)
- Photoshop
- VMware esxi
- Workbench
- Putty
- FileZilla
- GitHub
- OWASP ZAP, Legion
- Google chrome, Edge, Firefox
- Google Drive
- Word, PowerPoint, Excel
- Kali Linux
- Java Sdk

Automon

6.Securizacion

En el apartado de **securizacion**, realizamos una serie de comprobaciones en las cuales detallamos y corregimos en la maquina servidor y sitio web los problemas de seguridad.

Mediante diferentes políticas de seguridad, comandos y programas que utilicemos.

6.1 Securizacion del servidor

6.1.1_Credenciales de cuentas

A la hora de añadir más seguridad a los usuarios hemos decidido añadir contraseñas únicas para cada usuario.

Entendemos que esta aplicación es monousuaria, es decir que el empleado tendrá el mismo usuario para la aplicación móvil y web, solo tendrá un usuario.

Cada empleado tendrá su móvil y tendrá un sistema de autenticación para el caso de que terceras personas puedan acceder de forma no autorizada al dispositivo.

Las contraseñas deberán de tener al menos 8 caracteres entre ellos símbolos, mayúsculas, minúsculas...

Credenciales administradoras del sistema:

- Usuario administrador maquina servidor wordpress: root → Maristak2122
- Usuario root MySQL: root → %%my\$\$%ql

Credenciales empleadas de usuarios de la aplicación:

- Usuario empleado liher: liher124 → %L_%i_%h3RR1000
- Usuario empleado jon: jon1 → %J_%0_%n1000.
- Usuario administrador Admin: grupo2Admin → 5KjF1N2Qqk#v*WF3iC

6.1.2_Bloqueo puertos, Firewall

Para obtener mayor seguridad es posible **bloquear** los **puertos** que se sabe que pueden ser usados para atacar la red de la organización. Esto detiene servicios de red externa específicos. **Bloquear** los **puertos** puede proteger sus servicios más delicados. Cuando **bloqueamos** un **puerto**, se anulan todas las reglas en sus definiciones de políticas.

Para ello utilizamos la herramienta UFW, esta nos permite bloquear los puertos que no utilizamos y puedan crear una vulnerabilidad.

- Puertos bloqueados: 8080, 21
- Puertos abiertos: 389, 22, 80, 443, 3306

Bloquearemos los puertos que dejaremos de usar para obtener una mayor seguridad, ya que al no utilizarlos pueden ser una vía de entrada para los atacantes:

- `ufw status numbered`
- `ufw deny 21/tcp`
- `ufw deny 8080/tcp`
- `ufw reload`

```
root@2asir:~# ufw status numbered
Status: active

      To Action From
      --
[ 1] 389 ALLOW IN Anywhere
[ 2] 22 ALLOW IN Anywhere
[ 3] 80/tcp ALLOW IN Anywhere
[ 4] 3306/tcp ALLOW IN Anywhere
[ 5] OpenSSH ALLOW IN Anywhere
[ 6] 443/tcp ALLOW IN Anywhere
[ 7] 21/tcp DENY IN Anywhere
[ 8] 8080/tcp DENY IN Anywhere
[ 9] 389 (v6) ALLOW IN Anywhere (v6)
[10] 22 (v6) ALLOW IN Anywhere (v6)
[11] 80/tcp (v6) ALLOW IN Anywhere (v6)
[12] 3306/tcp (v6) ALLOW IN Anywhere (v6)
[13] OpenSSH (v6) ALLOW IN Anywhere (v6)
[14] 443/tcp (v6) ALLOW IN Anywhere (v6)
[15] 21/tcp (v6) DENY IN Anywhere (v6)
[16] 8080/tcp (v6) DENY IN Anywhere (v6)
```

6.1.3_Protocolo seguro y certificación (SSL y Https)

Añadiremos un soporte **SSL** al **servidor web** que permite establecer conexiones seguras y encriptadas entre el servidor y el cliente. De este modo, es posible cambiar contraseñas, con la certeza de que éstas no podrán ser interceptadas por terceros.

Mediante esta tecnología, los servidores pueden enviar tráfico de forma segura entre servidores y clientes sin la posibilidad de que los mensajes sean interceptados por terceros. El sistema de certificado también ayuda a los usuarios a verificar la identidad de los sitios con los que establecen conexión.

Creación de certificado auto firmado



Creamos el directorio donde se guardan los certificados ssl:
`/etc/apache2/ssl`

Y además también crearemos una subcarpeta con las clave pública y privada del certificado:
`/etc/apache2/ssl/grupo2Ciber/`

La TLS y la SSL funcionan utilizando una combinación de un certificado público y una clave privada. La clave SSL se mantiene secreta en el servidor. Se utiliza para cifrar contenido que se envía a los clientes.

El certificado SSL se comparte de forma pública con cualquiera que solicite el contenido. Puede utilizarse para descifrar el contenido firmado por la clave SSL asociada.

Creación del certificado auto firmado y las claves públicas y privadas:

```
openssl req -newkey rsa:2048 -x509 -nodes -days 365 -out
/etc/apache2/ssl/grupo2Ciber/grupo2Ciber.crt -
keyout /etc/apache2/ssl/grupo2Ciber/grupo2Ciber.key
```

Por último, configuraremos los Virtual Host que son una configuración muy útil al momento de desarrollar nuestras aplicaciones ya que las mismas permiten acceder fácilmente a una aplicación PHP alojada en un servidor web **Apache** a través de un dominio que configuremos en nuestro servidor.

Para ello, editar el archivo `/etc/apache2/sites-enabled/wordpress.conf`

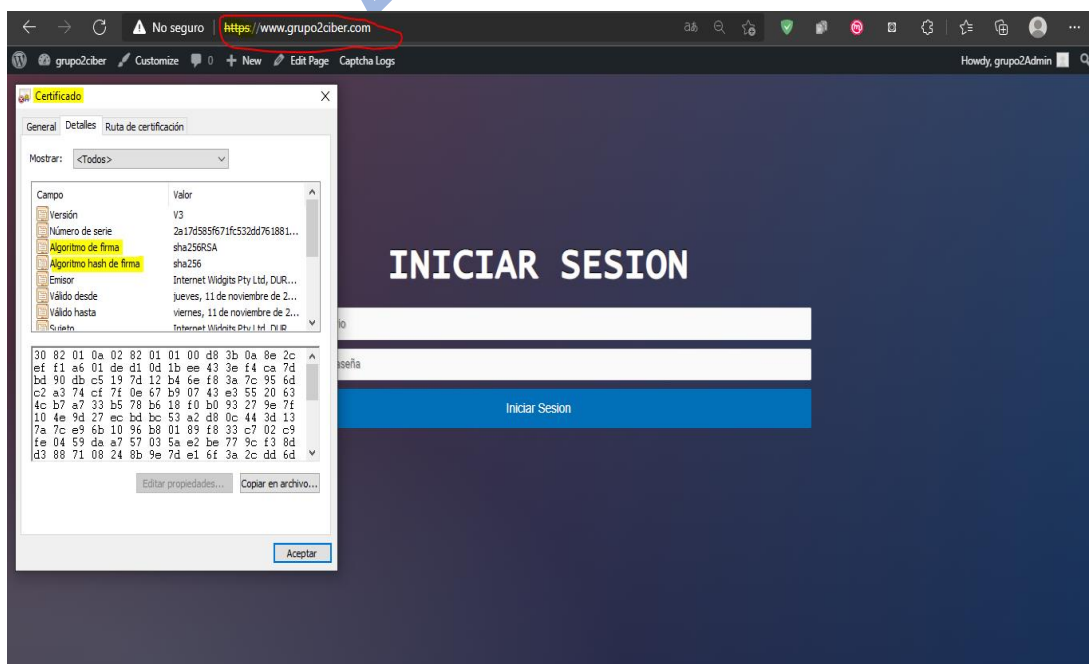
```
<VirtualHost *:80>
# RewriteEngine On
# RewriteCond %{HTTPS} !=on
# RewriteRule ^/?(.*) https://www.grupo2Ciber.com/$1 [R=301,L]

ServerName www.grupo2Ciber.com
Redirect permanent / https://www.grupo2Ciber.com/
ServerAdmin webmaster@localhost
DocumentRoot /var/www/wordpress
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:443>
ServerName www.grupo2Ciber.com
ServerAdmin webmaster@localhost
DocumentRoot /var/www/wordpress
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/grupo2Ciber/grupo2Ciber.crt
SSLCertificateKeyFile /etc/apache2/ssl/grupo2Ciber/grupo2Ciber.key
</VirtualHost>
```

La configuración anterior nos redirige del protocolo http al https, también nombramos los datos del servidor en el cual se aloja el sitio web “https://www.grupo2ciber.com”, además de poner la ubicación en la cual utilizara los certificados SSL.

En la siguiente **imagen** se observa, el protocolo utilizado en la pagina web, y el certificado.



6.1.3_Securizacion gestor de base de datos

Para obtener mayor seguridad aseguraremos el gestor de base de datos realizando el siguiente.

Mediante un **script** que se nos muestra durante la instalación, se nos pregunta que si queremos ponerle una contraseña al usuario root de MariaDB/MySQL le daremos que "SI".



Le añadiremos una contraseña nueva al usuario root de Mariadb/MySQL, además de realizar los siguientes pasos.

- Eliminar el usuario Anonymous
- Deshabilitar el acceso remoto de root
- Eliminar la base de datos test
- Recargar los privilegios

Utilizaremos el script mencionado anteriormente: `mysql_secure_installation`

```
Enter current password for root (enter for none):
OK, successfully used password, moving on... 1

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n 2
... skipping.

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y 3
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y 4
... Success!

By default, MySQL comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y 5
- Dropping test database...
ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist
... Failed! Not critical, keep moving...
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y 6
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!
```

El usuario root no podrá conectarse remotamente desde el cliente Workbench, para ello hemos creado un usuario administrador del sistema bdd y otro que solo tendrá acceso a la base de datos de wordpress.

- Usuarios BDD del sistema:

```
MariaDB [(none)]> SELECT user FROM mysql.user;
+-----+
| user          |
+-----+
| administrador3 |
| root          |
| user_wordpress |
+-----+
3 rows in set (0.000 sec)
```

- Permisos de usuarios del sistema:

```
MariaDB [(none)]> SHOW GRANTS FOR 'administrador3'@'%';
+-----+
| Grants for administrador3@%
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'administrador3'@`%` IDENTIFIED BY PASSWORD '*744D61DABFDEFFB24C9BA95C02EAA1132635A7EC' WITH GRANT OPTION |
+-----+
1 row in set (0.000 sec)
```

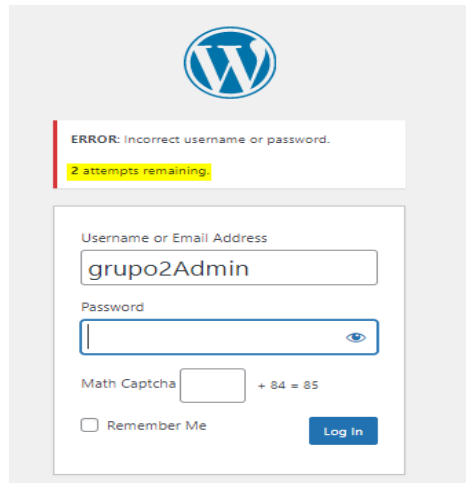
```
MariaDB [(none)]> SHOW GRANTS FOR 'user_wordpress'@'localhost';
+-----+
| Grants for user_wordpress@localhost
+-----+
| GRANT USAGE ON *.* TO 'user_wordpress'@`localhost` IDENTIFIED BY PASSWORD '*6604BC79504E6512561A3C5274819CF531413223' |
| GRANT ALL PRIVILEGES ON `wordpress`.* TO 'user_wordpress'@`localhost`
+-----+
2 rows in set (0.000 sec)
```

6.1.4_Securizacion gestor de contenidos (WordPress)

A la hora de **acceder** al panel de **administración** de wordpress, hemos añadido dos medidas de seguridad.

La primera es un captcha en la cual tengas hacer un tipo de suma, resta...

La segunda limita los intentos de sesión, en caso de pasarse tres veces cierra temporalmente el acceso a wordpress.



Comentarios:

Por la parte de los comentarios están desactivados debido a los siguientes motivos:

- Para evitar Spam
- Apariencia profesional
- La página no está diseñada para generar conversaciones
- Para aumentar el rendimiento

Paginas:

En las páginas, hemos añadido código PHP en el cual no permite acceder a ellas si no has iniciado sesión el sitio web y te redirige al login principal.

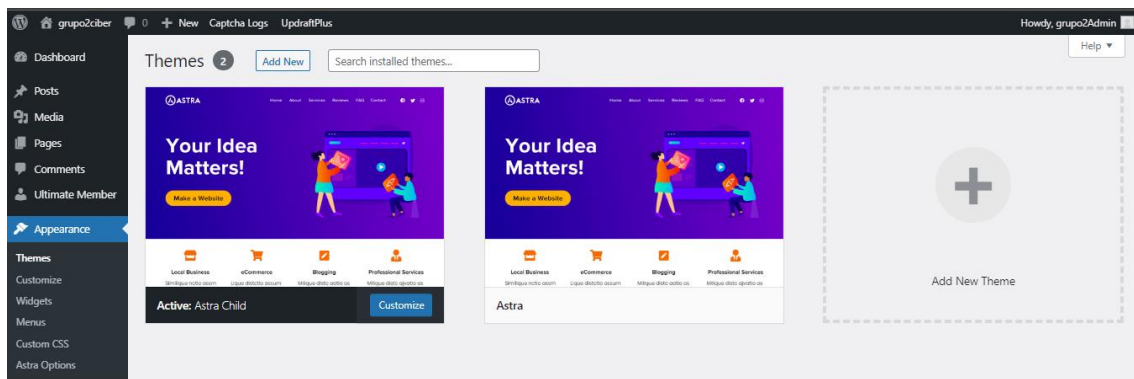
Temas:

A la hora de ofrecer más seguridad a los tema, tenemos que tener en cuenta los siguientes puntos.

- Usar un buen theme, bien programado y de confianza, Astra en nuestro caso.
- Tener las últimas versiones de los themes almacenado en nuestro WordPress (están activos o no).
- Solo usamos un theme, quitar del servidor el resto.
- Usar un theme hijo y del tema principal.

Muestra de los temas **no utilizados** estén borrados:

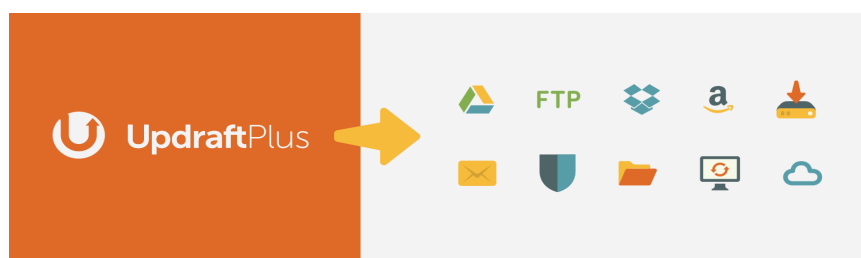
```
root@2asir:~# ls /var/www/wordpress/wp-content/themes/  
astra astra-child index.php
```



Copia de seguridad:

En caso de que podamos perder el contenido por una razón u otra tendremos que realizar copias periódicas de seguridad tanto de los ficheros como de la base de datos de nuestro WordPress.

- Para ello utilizaremos el plugin: UpdraftPlus



Cambio de prefijo de las tablas

WordPress tiene algunas opciones que puedes considerar para mejorar la seguridad de tu sitio, una de estas opciones es el **cambio de prefijo de las tablas de la base de datos**, si bien no es una solución infalible para el ataque de inyección SQL, dificultará la acción de un atacante.



Servidor de la base de datos: localhost

Prefijo de tabla: wp_

Enviar

Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona.

Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

¿Qué prefijo usar?

Es importante tener en cuenta que debido a que estas cambiando el nombre de las tablas, sólo deberías usar caracteres alfanuméricos aleatorios y guion bajo, por ejemplo: **mkdwp_**

Por nuestro criterio el prefijo que añadiremos será el siguiente: **mkdwp_**

Cambiar el archivo wp-config.php

Para ello, acceder al archivo *wp-config.php* en la raíz del el sitio web, y luego ubicar la variable *table_prefix* y cámbialar

- `$table_prefix = 'mkdwp_';`

```
GNU nano 3.2 /var/www/wordpress/wp-config.php

/**
 * WordPress database table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'mkdwp_';

/**
 * For developers: WordPress debugging mode.
 */
```

6.1.5_Securizacion ficheros y directorios

Permisos de carpetas y ficheros:

Los permisos son atributos que se definen para los directorios y ficheros de WordPress y que permiten ejecutarlos, leerlos o modificarlos. Los **permisos** que usaremos serán los siguientes:

- Ficheros: 644
- Directorios: 755
- .htaccess: 644 o 604
- wp-config.php: 644 o 604

Una vez vistos los permisos, eliminaremos los ficheros para una mayor seguridad: `readme.html`, `license.txt`, `wp-config-sample.php` y `/wp-admin/install.php`

De esta forma, los permisos generales para nuestra instalación de WordPress son los siguientes:

Permisos en archivos y directorios

- **Directorios** → `find /var/www/wordpress/ -type d -exec chmod 755`
- **Scripts** → `find /var/www/wordpress/ -type f -exec chmod 644`

1	<code>index.php</code>	644
2	<code>licencia.txt</code>	644
3	<code>license.txt</code>	644
4	<code>readme.html</code>	644
5	<code>wp-activate.php</code>	644
6	<code>wp-admin</code>	755
7	<code>wp-blog-header.php</code>	644
8	<code>wp-comments-post.php</code>	644
9	<code>wp-config-sample.php</code>	644
10	<code>wp-content</code>	755
11	<code>wp-cron.php</code>	644
12	<code>wp-includes</code>	755
13	<code>wp-links-opml.php</code>	644
14	<code>wp-load.php</code>	644
15	<code>wp-login.php</code>	644
16	<code>wp-mail.php</code>	644
17	<code>wp-settings.php</code>	644
18	<code>wp-signup.php</code>	644
19	<code>wp-trackback.php</code>	644
20	<code>xmlrpc.php</code>	644
21	<code>.htaccess</code>	644

Comando realizado para cambiar los permisos:

```
root@2asir:/var/www/wordpress# find /var/www/wordpress/ -type d -exec chmod 755 {} \;
root@2asir:/var/www/wordpress# find /var/www/wordpress/ -type f -exec chmod 644 {} \;
root@2asir:/var/www/wordpress#
```

Medidas de seguridad en el fichero .htaccess

.htaccess es un archivo de configuración en servidores Apache que permite aplicar distintas políticas de accesos a directorios y archivos para adicionar medidas de seguridad.

Evitar la navegación por directorios

```
Options All -Indexes
```

Desactivar los métodos Trace/Track

```
RewriteEngine On
RewriteBase /
RewriteCond %{QUERY_STRING} (author=\d+) [NC]
RewriteRule .* - [F]
```

Denegar el acceso al Xmlrpc.php, wp-config.php y error_log.php

```
<Files xmlrpc.php>
order deny,allow
deny from all
</Files>
```

```
<files wp-config.php>
order allow,deny
deny from all
</files>
```

```
<files error_log>
Order allow,deny
Deny from all
</files>
```

Tres encabezados X-Security

Contra ataques XSS, click-jacking y rastreo de contenido

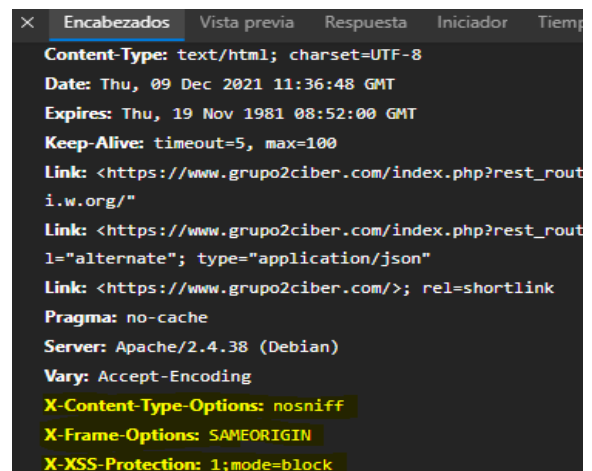
```
<IfModule mod_headers.c>

    Header set X-XSS-Protection "1; mode=block"
    Header always append X-Frame-Options SAMEORIGIN
    Header set X-Content-Type-Options nosniff
    Header set Referrer-Policy: no-referrer
    Header set Feature-Policy: "geolocation none"
    Header set Strict-Transport-Security "max-
age=31536000;includeSubDomain$

</IfModule>
```

No mostrar la versión del servidor

```
ServerSignature Off
```



```

Content-Type: text/html; charset=UTF-8
Date: Thu, 09 Dec 2021 11:36:48 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
Link: <https://www.grupo2ciber.com/index.php?rest_route=/wp-json/wp/v2/users/>
Link: <https://www.grupo2ciber.com/index.php?rest_route=/wp-json/wp/v2/users/>
Link: <https://www.grupo2ciber.com/>; rel=shortlink
Pragma: no-cache
Server: Apache/2.4.38 (Debian)
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1;mode=block
  
```

Medidas de seguridad wp-config.php

El archivo **wp-config.php** almacena datos como los detalles de la conexión a la base de datos, tablas de prefijos, vías a directorios específicos y muchas opciones relacionadas

```
/* Desactivar cron wordpress */  
define('DISABLE_WP_CRON', true);
```

```
/* Habilitar la configuración segura de cookies con HTTPOnly */  
@ini_set('session.cookie_httponly', true);  
@ini_set('session.cookie_secure', true);  
@ini_set('session.use_only_cookies', true);
```

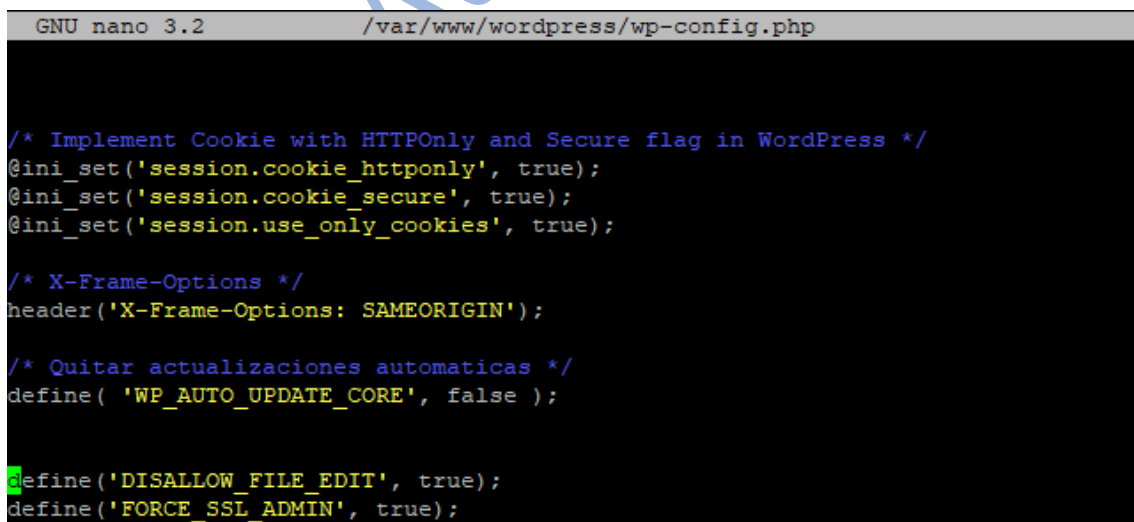
```
/* Deshabilitar la depuración */  
define('WP_DEBUG', false);
```

```
/* Forzando el login seguro con SSL */  
define('FORCE_SSL_ADMIN', true);
```

```
/* Desactivar la opción de editar ficheros desde wp-admin */  
define('DISALLOW_FILE_EDIT', true);
```

```
/* Desactivar actualizaciones automáticas */  
define('WP_AUTO_UPDATE_CORE', false);
```

Foto en la cual se muestra la **configuración** del fichero:



```
GNU nano 3.2 /var/www/wordpress/wp-config.php  
  
/* Implement Cookie with HTTPOnly and Secure flag in WordPress */  
@ini_set('session.cookie_httponly', true);  
@ini_set('session.cookie_secure', true);  
@ini_set('session.use_only_cookies', true);  
  
/* X-Frame-Options */  
header('X-Frame-Options: SAMEORIGIN');  
  
/* Quitar actualizaciones automaticas */  
define('WP_AUTO_UPDATE_CORE', false);  
  
define('DISALLOW_FILE_EDIT', true);  
define('FORCE_SSL_ADMIN', true);
```

Medidas de seguridad en functions.php

functions.php es el "plugin de la plantilla" en otras palabras a través de él podemos añadir funcionalidades extra (tal y como hacen los plugins) a nuestra plantilla. Para añadir estas funciones extras se utiliza lenguaje PHP. Se utiliza para lanzar hooks y filtros predefinidos en WordPress que permiten modificar su funcionamiento.

Borrar versión wordpress

```
remove_action('wp_head', 'wp_generator');  
add_filter('the_generator', '__return_false');
```

Foto en la cual se muestra la **configuración** del fichero:

```
/var/www/wordpress/wp-content/themes/astra-child/functions.php  
  
endif;  
add_action( 'wp_enqueue_scripts', 'child_theme_configurator_css', 10 );  
  
// END ENQUEUE PARENT ACTION  
  
/* Borrar version wordpress */  
remove_action('wp_head', 'wp_generator');  
add_filter('the_generator', '__return_false');  
  
/* Desactivar X-frame */  
add_action( 'send_headers', 'add_header_seguridad' );  
function add_header_seguridad() {  
header( 'X-Content-Type-Options: nosniff' );  
header( 'X-Frame-Options: SAMEORIGIN' );  
header( 'X-XSS-Protection: 1;mode=block' );  
}  
  
/* Cómo deshabilitar el API con código */  
add_filter( 'rest_authentication_errors', function( $result ) {
```

6.2 Validación

A la hora de realizar la **validación** seguiremos las recomendaciones **OWASP**, por los siguientes motivos.

OWASP establece y explica las diez vulnerabilidades más importantes que pueden aparecer en un sitio web.

Los atacantes pueden usar diferentes rutas a través de la aplicación de un negocio para causar importantes daños al mismo.

El **riesgo total** para una empresa viene dado por la unión de:

- La probabilidad asociada con cada agente de amenaza
- Vector de ataque
- Debilidad de seguridad
- Estimación del impacto técnico
- Estimación del impacto para el negocio

Cada aplicación y cada empresa son diferentes por lo que habrá que **evaluar** el **riesgo** en cada caso enfocándonos en:

- Agentes amenazantes
- Controles de seguridad
- Impacto de negocio

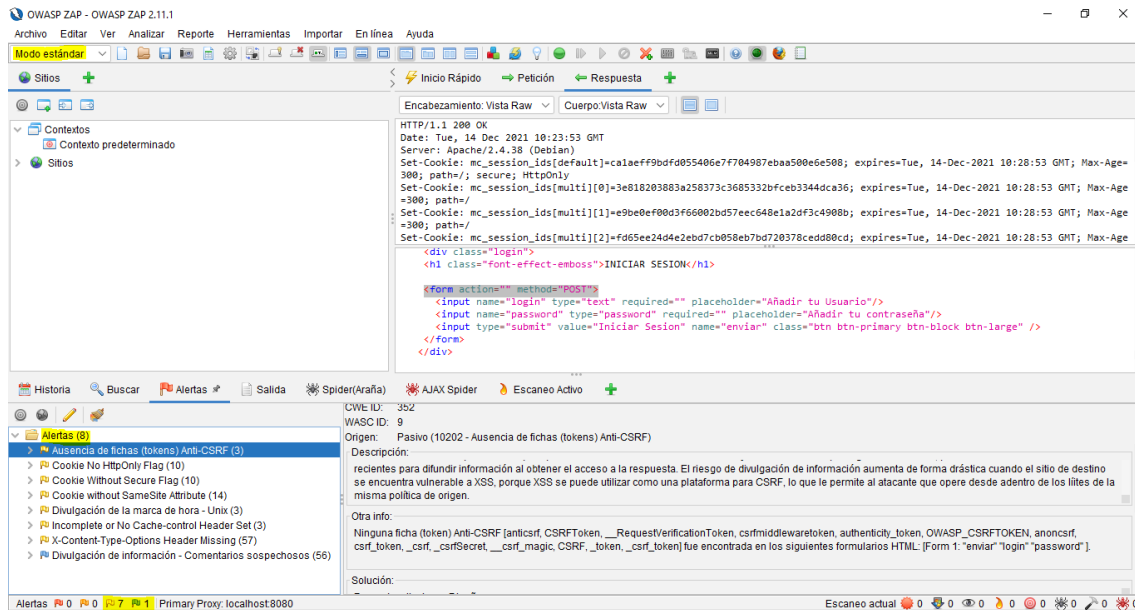
Las **diez vulnerabilidades** más importantes y comunes de las aplicaciones web son las siguientes:

- Inyección SQL
- Secuencia de comandos en sitios cruzados (XSS)
- Pérdida de autenticación y gestión de sesiones
- Referencia directa insegura a objetos
- Falsificación de peticiones en sitios cruzados
- Configuración de seguridad defectuosa
- Almacenamiento criptográfico inseguro
- Fallo de restricción de acceso a URL
- Protección insuficiente en la capa de transporte
- Redirecciones y reenvíos no validados

Una vez **definidas** las **vulnerabilidades** más importantes, mediante el uso de la aplicación OWASP ZAP, identificaremos las vulnerabilidades haciendo pruebas de penetración en la página web, estas una vez terminadas nos mostraran información acerca de estas.

Mediante la **herramienta** realizamos un test de penetración en modo estándar para comprobar las vulnerabilidades existentes y corregirlas posteriormente.

Test realizado:



Como se poder ver en la **imagen** anterior, una vez finalizada la prueba podemos ver que existen **8 vulnerabilidades**, pero lo que se indica de ellas son **“low”** leves, esto quiere decir que no son de gravedad alta y no son peligrosas.

Este **test** esta realizado ya con el **sitio web** finalizado y securizado, anteriormente cuando en las primeras fases la página tenía errores graves los cuales hemos ido securizando.

Por ejemplo: al principio cualquier persona podía acceder a la página gestión de gastos sin iniciar sesión solo con poner la “URL”, para securizarlo añadimos código PHP para que solo pudieran acceder los usuarios que iniciaran sesión a la web.

Validación mediante legión

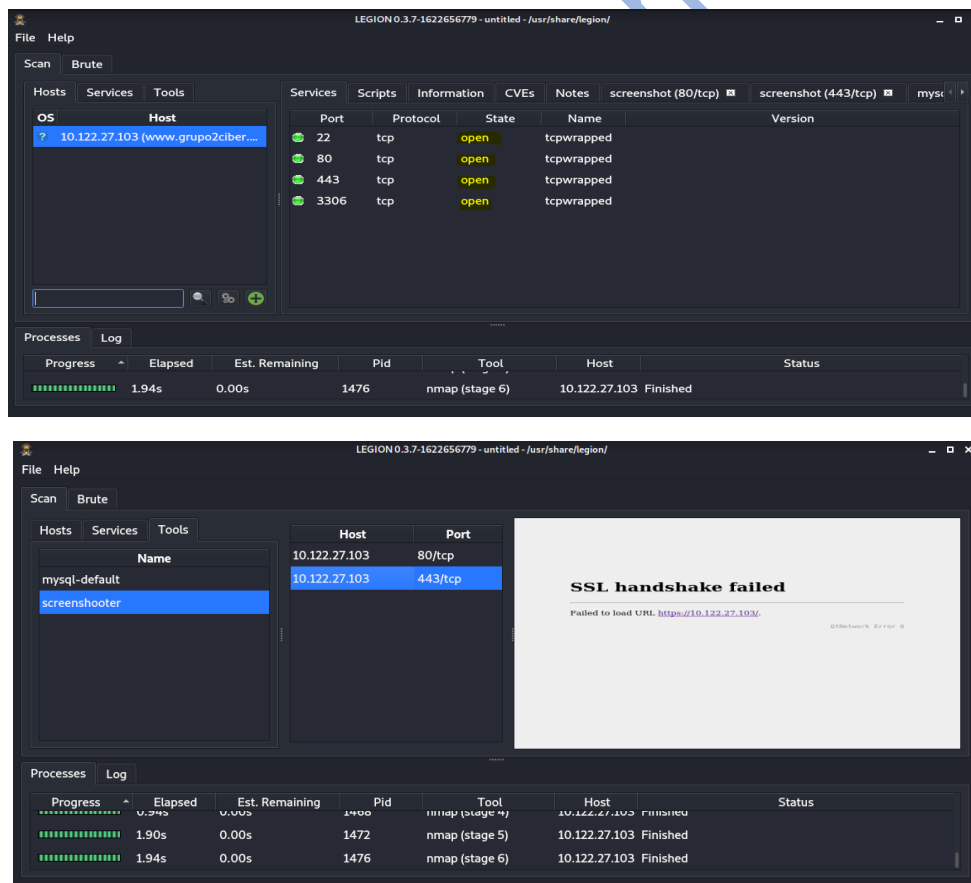
Mediante la herramienta “LEGION”, permitirá realizar las pruebas de penetración. Usando esto podemos hacer un escaneo automático y encontrar vulnerabilidades en el sitio web.

Para realizar la prueba primero escaneará la IP “10.122.27.103” o la dirección web con nmap y luego ejecutará Nikto en la IP o dirección web de destino.

Legión probará con varias herramientas automatizadas como Shodan, whataweb, Vulners, Hydra, SMBenum, dirbuster, sslyzer, webslayer y más (con casi 100 scripts programados automáticamente).

Esta escaneara automáticamente para encontrar vulnerabilidades en el sitio web.

Test realizado:



Como se puede observar los puertos que están abiertos son los esenciales para el funcionamiento del sitio web y no se observan otros puertos abiertos, a la hora de acceder la aplicación tampoco puede por http ya que esta redireccionado a https, los aspectos más importantes están securizados, como muestra la imagen anterior.

6.3 Mejoras API REST banco

Especificaciones de la API REST actual:

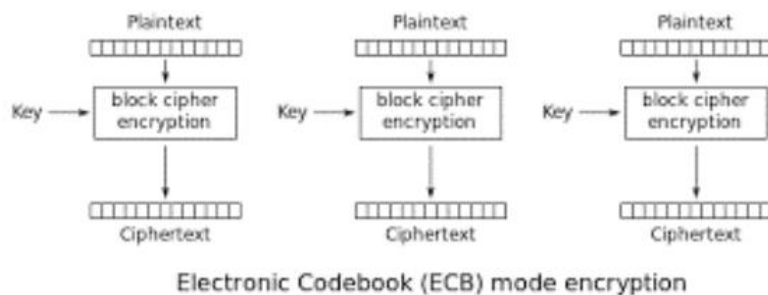
Especificaciones de seguridad	
Algoritmo	AES-256
Modo	ECB
Clave	Solicitar al cliente

Mejoras que indicamos para mejorar la seguridad del api rest del banco, principalmente en dos apartados:

Diferencias entre ECB y CBC:

1.ECB:

Entre el modo ECB y el modo CBC, siempre es mejor elegir el modo CBC. El modo BCE filtra información sobre el texto plano porque los bloques de texto plano idénticos producen bloques de texto cifrado idénticos. Un texto cifrado nunca debe filtrar ninguna información sobre el texto plano utilizado para crearlo, por lo que el modo BCE es inseguro y nunca debe usarse.



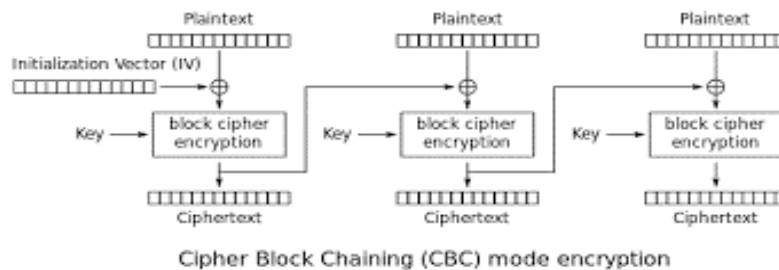
2.CBC:

El modo CBC, por otro lado, es uno de los modos de operación de cifrado de bloques más utilizados debido a su facilidad de implementación y soporte para el descifrado paralelizado.

Los problemas del modo ECB surgen del hecho de que cada bloque del texto plano se cifra de forma completamente independiente. El modo CBC elimina este problema al llevar información del cifrado o descifrado de un bloque al siguiente.

Este vector de inicialización "IV" utiliza el texto cifrado resultante se utiliza para llevar la información al cifrado del siguiente bloque y así sucesivamente.

Esta relación entre bloques ayuda a proteger contra bloques de texto plano idénticos que producen bloques de texto cifrado idénticos. Dado que cada bloque del texto plano es XORed con un IV diferente antes del cifrado, produce un texto cifrado único.



Securización del protocolo de Transferencia de Hipertexto:

Actualmente el **protocolo** que utiliza el API REST del banco es http y proponemos cambiar al protocolo https por las siguientes razones.

- **Cifrado:** Los datos del enviado y recibidos, entre el usuario y el servidor, viajan encriptados (y no en modo texto).
- **Integridad de datos:** Los datos no pueden modificarse durante su viaje de origen a destino, sin que el protocolo lo detecte.
- **Autenticación:** Demuestra que el usuario se está comunicando con la página Web que está visitando, y no hay ningún tipo de suplantación. Es decir, el usuario tiene la certeza de estar visitando la página, que quiere visitar.

7. Información entregada

En el momento de la entrega de la aplicación se hará entrega de la siguiente archivos y documentos.

- Presentación del proyecto está en la url: [Presentación Grupo2](#)
- Una carpeta que contiene diferentes documentos que hemos usado durante la creación del proyecto: [Carpeta Grupo2](#)
- Planificación del proyecto: [Planificación grupo2](#)
- GitHub código wordpress y aplicación Android: [Archivos aplicación Android y web](#)
- Los archivos **Apk-debug** que son los ejecutables de la aplicación: [Aplicacion-multiplataforma-grupo2/Android at main · Liher124/Aplicacion-multiplataforma-grupo2 \(github.com\)](#)
- Servidor: [VMware ESXi servidor](#)
 - ciber2/KWjgOW4T
 - root/Maristak2122

8.Conclusion

Principalmente esta todo lo pedido por el cliente, además también las especificaciones técnicas han sido superadas y nuestros objetivos han sido cumplidos con éxito.

Problemas o dudas a la hora de realizar el proyecto que nos hemos encontrado:

- Cómo hacer uso del protocolo https y SSL a la hora de securizar la web de wordpress.
- Cómo hacer uso para consumir servicios API REST del banco.
- Cómo enlazar las bases de datos Mariadb y SQLite. --> No realizado.
- Cómo incluir el código PHP en el sitio wordpress.
- Cómo securizar la aplicación Android.
- Cómo introducir las horas ya que en la base de datos se introducen primero con el año, y en el código PHP primero el día.
- Cómo incluir imágenes en la base de datos.

Finalmente nos hemos podido cumplir el objetivo de **sincronizar** las bases de datos y la introducción de la imagen de los tiques lo cual, funcionaran la base de datos de manera local, por lo demás todo se ha realizado con éxito.

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

9.Referencias

API REST

[Cómo consumir un Webservice REST con PHP - Leeway Academy \(leewayweb.com\)](#)

[Cómo consumir API Rest con PHP de forma sencilla \(codigonaranja.com\)](#)

☒ [API/Rest en Android Studio TUTORIAL 【 2021 】 \(cursoandroidgratis.com.es\)](#)

[Cómo usar la API REST en Android \(biblioteca Volley\) \(ichi.pro\)](#)

OWASP ZAP

[OWASP - Seguridad en la web | Ciberseguridad](#)

[Tutorial de OWASP ZAP: Revisión completa de la herramienta OWASP ZAP - Otro \(myservername.com\)](#)

HTACCESS, CONFIG.PHP, FUNCTIONS.PHP ...

[Seguridad en Wordpress, top 10 consejos \(aurea.es\)](#)

[Tutorial del Servidor Apache HTTP: Ficheros .htaccess - Servidor HTTP Apache Versión 2.4](#)

[How To Secure Your WordPress Site With WP-Config.php? - MalCare](#)

SSL, HTTPS

[Cómo crear un certificado SSL autofirmado para Apache en Ubuntu 18.04 | DigitalOcean](#)

[Cómo configurar SSL en Apache \(Debian\) \(linuxito.com\)](#)

UFW, PORTS

[Cómo configurar un firewall con UFW en Debian 9 | DigitalOcean](#)

[How to Check for Open Ports on Debian 10 – VITUX](#)

[How to check if port is in use on Linux or Unix - nixCraft \(cyberciti.biz\)](#)

Mariadb, SQLite

[How To Install MariaDB on Debian 10 | DigitalOcean](#)

[Proyecto ejemplo de App Android con bbdd SQLite – Academia Android](#)

[Cómo guardar datos con SQLite | Desarrolladores de Android | Android Developers](#)

[Crear base de datos SQLite \(w3big.com\)](#)