1. Identificación de Activos

Los activos se clasifican en cinco grandes grupos: *hardware, software, información, servicios, personal y comunicaciones*.

1.1 Activos Hardware

Activo	Descripción	Importancia
Equipo servidor local (El portátil)	Donde se aloja XAMPP, Apache, MySQL y todo el proyecto.	Alto
Router	Proporciona conexión DHCP y acceso a la red.	Medio
Disco duro del servidor	Almacena la aplicación y la base de datos.	Alto
Periféricos básicos (teclado, ratón, pantalla)	Necesarios para la administración del sistema.	Bajo

---

1.2 Activos Software

Activo	Descripción	Importancia
Sistema operativo Windows	Plataforma donde se ejecutan los servicios.	Alto
XAMPP (Apache + MySQL + PHP)	Infraestructura crítica para que la aplicación funcione.	Alto
Aplicación web "Gestor de Reservas"	Código fuente PHP, HTML, CSS.	Muy alto
Script Python para informes	Genera reportes en PDF.	Medio
PHPMailer	Envío de correos al cliente.	Medio
Navegador web del usuario	Medio de acceso al servicio.	Medio
Git y GitHub	Gestión del código y copias externas.	Alto

1.3 Activos de Información

Activo	Contenido	Importancia
Base de datos gestion_reservas	Información crítica sobre clientes, usuarios, reservas, estados, precios.	Muy alto
Credenciales de acceso	Usuario empleado, clientes y contraseñas.	Muy alto
Logs de la aplicación y Apache	Información que puede ser necesitada.	Medio
Informes PDF generados en Python	Datos de negocio procesados.	Medio
Imágenes de los vehículos	Información visual usada por la aplicación.	Bajo

---

1.4 Activos de Servicios

Servicio	Descripción	Importancia
Servidor Web Apache	Publica el portal del rent-a-car.	Muy alto
Servidor MySQL	Motor de la base de datos del sistema.	Muy alto
Servicio DHCP del router	Asigna IP dinámica al servidor en entorno real.	Medio
Resolución DNS/Hosts	Permite acceso mediante rentacar.local	Medio
Servicio de correo SMTP	Envío automático de confirmaciones.	Alto
GitHub (almacenamiento remoto del código)	Copias de seguridad y versionado.	Alto

---

1.5 Activos Humanos

Activo	Rol	Importancia
Administrador / Empleado	Valida reservas, gestiona clientes y vehículos.	Muy alto
Cliente del rent-a-car	Usa la web para crear reservas.	Alto
Desarrollador del sistema	Administra el servidor, corrige código, aplica parches.	Muy alto
Tutor / evaluador	Verifica funcionamiento.	Medio

1.6 Activos de Comunicaciones

Activo	Descripción	Importancia
Conexión HTTP (puerto 80)	Acceso principal a la aplicación.	Muy alto
Conexión MySQL (3306)	Acceso a la base de datos desde la aplicación.	Alto
Red local del centro / casa	Permite conectividad del proyecto.	Alto
Canal SMTP externo	Envía correos de confirmación.	Alto

Resumen de activos críticos

Estos son los activos más importantes que deben protegerse especialmente:

1. **Base de datos gestion_reservas.**
2. **Código de la aplicación (PHP + Python).**
3. **Servidor Apache + MySQL.**
4. **Credenciales de usuario y contraseñas.**
5. **Dominio interno rentacar.local.**
6. **Copia en GitHub (código fuente).**

2. Análisis de Amenazas del Sistema

A continuación se identifican las amenazas que pueden afectar a los activos del sistema gestor de reservas *Autos Costa Sol*.
Se clasifican por categorías: amenazas físicas, lógicas, humanas, operativas y externas.

2.1 Amenazas sobre Activos Hardware

Activo	Amenaza	Descripción

PC servidor	***Fallo de hardware***	**Sobrecalentamiento, fuente de alimentación dañada, disco defectuoso.**
	Pérdida de energía	**Un corte eléctrico interrumpe Apache/MySQL provocando pérdida de datos en memoria.**
	Robo o acceso físico no autorizado	**Alguien podría acceder al equipo y copiar datos.**
Disco duro	***Corrupción de datos***	**Sectores dañados o fallo mecánico.**
	Eliminación accidental	**Borrado involuntario de la carpeta del proyecto o base de datos.**
Router / red local	***Fallo del router***	**Deja inaccesible el servicio.**
	Asignación incorrecta por DHCP	**El PC servidor recibe otra IP y se pierde el acceso por DNS.**

2.2 Amenazas sobre Activos Software

Activo	Amenaza	Descripción
Sistema operativo Windows	***Malware / virus***	**Riesgo por ejecutar código o software descargado.**
	Actualizaciones automáticas	**Un reinicio inesperado detiene los servicios.**

XAMPP (Apache + MySQL)	***No disponibilidad del servicio***	**Apache o MySQL dejan de funcionar.**
	Configuración incorrecta	**Cambios en VirtualHost o php.ini pueden impedir acceso.**
	Ataques web	**Inyección SQL, XSS, acceso no autorizado.**
Aplicación PHP	***Errores de programación***	**Formularios mal validados, path traversal, etc.**
	Falta de parches	**Vulnerabilidades no corregidas.**
Python + Reportlab	***Bloqueo al generar PDF***	**Error al acceder a archivos o permisos.**
PHPMailer	***Fallo de envío***	**Servidor SMTP caído o credenciales incorrectas.**

2.3 Amenazas sobre Activos de Información

Activo	Amenaza	Descripción
Base de datos de reservas	***Pérdida total de datos***	**Formateo accidental, borrado manual o fallo SQL.**
	Acceso no autorizado	**Un atacante obtiene reservas o datos personales.**

	Modificación indebida	**Cambios en precios, vehículos o estados de reserva.**
	Exposición de credenciales	**Robo de contraseñas hash o sesión de usuario.**
Credenciales y sesiones	***Robo o suplantación***	**Cookies robadas, contraseñas débiles.**
Imágenes y archivos	***Manipulación no autorizada***	**Sustitución de imágenes o archivos de la web.**

2.4 Amenazas sobre Servicios

Servicio	Amenaza	Descripción
Apache	***DDoS local***	**Exceso de peticiones o errores causa caída.**
	Configuración insegura	**Directorios accesibles, listados habilitados.**
MySQL	***Inyección SQL***	**Riesgo directo sobre reservas y usuarios.**
	Exceso de conexiones	**Servicio se satura.**
DNS / Hosts	***Conflictos de nombres***	**Otro servicio usando rentacar.local.**
	Modificación maliciosa	**Alteración del archivo hosts.**

SMTP	***Bloqueo por proveedor***	**Envíos masivos → email marcado como spam.**
GitHub	***Exposición del código***	**Subir credenciales o configuración sensible.**

2.5 Amenazas Humanas

Actor	Amenaza	Descripción
Cliente	***Uso incorrecto del sistema***	**Formularios mal completados o repetidos.**
Empleado	***Error en gestión***	**Confirmar o cancelar reservas por error.**
	Acceso indebido	**Empleado actuando como cliente o viceversa.**
Administrador (tú)	***Borrado accidental***	**Eliminación de datos o carpetas sin backup.**
Usuarios externos	***Ataques maliciosos***	**Intentos de login, scans, fuerza bruta.**

2.6 Amenazas sobre Comunicaciones

Activo	Amenaza	Descripción

HTTP sin cifrar	***Sniffing de datos***	**Robo de credenciales en red local (teórica).**
Conexión MySQL	***Interceptación***	**Captura de tráfico SQL si se expusiera.**
Red local	***Fallo de conectividad***	**Se corta el acceso a rentacar.local.**

Resumen General de Amenazas Detectadas

Las amenazas más significativas para el proyecto son:

Críticas:

- **Inyección SQL.**
- **Pérdida o corrupción de la base de datos.**
- **Accesos no autorizados (clientes/empleados).**
- **Caída de servicios Apache o MySQL.**
- **Exposición o fuga de información personal (RGPD).**

Altas:

- **Borrado accidental de datos.**
- **Fallo del equipo servidor.**
- **Malware o ransomware.**
- **Errores de configuración en XAMPP.**

Medias/Bajas:

- **Manipulación de imágenes.**
- **Problemas con DNS local.**

- **Fallo del generador de PDF.**

---

3. Análisis de Vulnerabilidades del Sistema

Las vulnerabilidades son debilidades del sistema que podrían ser explotadas por amenazas. A continuación se identifican las principales vulnerabilidades.

3.1 Vulnerabilidades en Hardware

Activo	Vulnerabilidad	Descripción
PC servidor local	***Sin SAI ni protección eléctrica***	**Un corte eléctrico puede apagar el sistema y provocar corrupción en MySQL.**
	Acceso físico no controlado	**Cualquier usuario que acceda al PC puede copiar la base de datos.**
Disco duro	***Sin RAID ni redundancia***	**Si falla el disco, se pierde toda la información.**
Router	***Configuración por defecto***	**Contraseñas por defecto pueden comprometer la red local.**

3.2 Vulnerabilidades en Software

Componente	Vulnerabilidad	Descripción
Windows	***Sin hardening***	**Servicios innecesarios habilitados o firewall mal configurado.**
XAMPP	***Entorno de desarrollo poco seguro***	**Apache/MySQL no están pensados para producción.**
	Puertos abiertos innecesariamente	**Riesgo si el PC se expone a Internet.**
Apache	***No usa HTTPS***	**El tráfico viaja sin cifrado.**
	Directory Listing si no se configura	**Alguien podría listar carpetas.**
MySQL	***Contraseña del root por defecto***	**Facilita accesos no autorizados.**
	Inyección SQL	**Formularios que no validan correctamente los datos.**
PHP (aplicación)	***Validación insuficiente***	**Riesgo de XSS, SQLi, CSRF.**
	Errores visibles	**Mostrar errores puede filtrar rutas o detalles técnicos.**

PHPMailer	***SMTP sin cifrado TLS***	**Las credenciales del correo podrían filtrarse.**
Python	***Permisos de escritura en PDF***	**Cualquiera podría sobrescribir el informe.**

3.3 Vulnerabilidades en Información

Activo	Vulnerabilidad	Descripción
Base de datos	***Sin cifrado***	**Datos personales almacenados en texto legible.**
	Sin copias de seguridad regulares	**Riesgo alto de pérdida de datos.**
	Acceso local sin restricción por firewall	**MySQL responde a cualquier petición interna.**
Credenciales de usuarios	***Contraseñas débiles***	**Si un usuario usa una contraseña simple, es fácilmente atacable.**
	Sesiones sin duración limitada	**Riesgo de secuestro de sesión.**
Archivos e imágenes	***Falta de control de integridad***	**Pueden ser modificados sin dejar rastro.**

3.4 Vulnerabilidades en Servicios

Servicio	Vulnerabilidad	Descripción
DNS local (hosts)	***Alteración del archivo hosts***	**Podría redirigir tráfico a otro sitio.**
DHCP	***Dependencia de IP dinámica***	**Si cambia la IP del servidor, DNS deja de funcionar.**
SMTP	***Configuración insegura***	**Permite ataques de spoofing si no usa autenticación correcta.**
GitHub	***Subida accidental de datos sensibles***	**Riesgo al subir `.env`, contraseñas o configuraciones.**

3.5 Vulnerabilidades Humanas

Actor	Vulnerabilidad	Descripción
Clientes	***Desconocimiento de seguridad***	**Pueden usar contraseñas débiles o compartir la cuenta.**
Empleado	***Errores de gestión***	**Confirmar o cancelar reservas erróneamente.**
	Falta de formación	**Riesgo al manipular el sistema sin conocimientos técnicos.**

Administrador	***Modificación accidental de archivos***	**Un cambio en XAMPP o Apache puede tirar el sistema.**
	Falta de experiencia en ciberseguridad	**puede dejar configuraciones por defecto.**

3.6 Vulnerabilidades de Comunicaciones

Activo	Vulnerabilidad	Descripción
HTTP (puerto 80)	***Sin cifrado TLS***	**Contraseñas viajan en texto plano.**
MySQL	***Escucha en localhost sin restricción***	**Si se cambia por error, puede abrirse a la red.**
Red local	***No segmentada***	**Cualquier equipo puede intentar conectarse al servicio.**

Resumen de vulnerabilidades críticas

Las más importantes que deben solucionarse:

Críticas:

- **Inyección SQL por falta de validación.**
- **MySQL sin cifrado ni firewall.**
- **Apache sin HTTPS.**

- **Sin backups periódicos de la base de datos.**
- **Posible alteración de `hosts` o VirtualHost.**
- **SMTP sin cifrado (PHPMailer).**

Importantes:

- **Contraseñas débiles.**
- **Errores humanos en gestión de reservas.**
- **Falta de autenticación multifactor.**

4. Matriz de Riesgos del Sistema

A continuación se presenta la matriz de riesgos resultante del cruce entre amenazas y vulnerabilidades detectadas.

Leyenda de niveles de riesgo

Nivel	Color	Descripción
Crítico	🔴 Rojo	Riesgo inaceptable: requiere mitigación inmediata
Alto	🟠 Naranja	Riesgo significativo: requiere medidas correctoras
Medio	🟡 Amarillo	Riesgo moderado: debe controlarse
Bajo	🟢 Verde	Riesgo aceptable con monitorización mínima

Matriz completa de riesgos

Riesgo identificado	Probabilidad	Impacto	Nivel de riesgo	Comentario
Pérdida total de la base de datos	Media	Alto	🔴 Crítico	Sin backups automáticos.
Inyección SQL	Alta	Alto	🔴 Crítico	Formularios manipulables si no se validan correctamente.
Caída del servicio Apache/MySQL	Media	Alto	🔴 Crítico	Afecta a todo el sistema.
Acceso no autorizado de un atacante	Media	Alto	🔴 Crítico	HTTP sin cifrado, sesiones largas.
Exposición de credenciales SMTP o BD	Media	Alto	🔴 Crítico	Riesgo real si se suben archivos sensibles a GitHub.
Fallo del disco del servidor	Baja	Alto	🟠 Alto	El equipo no tiene redundancia.
Errores humanos del empleado	Alta	Medio	🟠 Alto	Confirmación/cancelación de reservas incorrecta.
Correo no enviado o falla SMTP	Media	Medio	🟠 Alto	El cliente podría no recibir confirmación.
Modificación del archivo hosts	Baja	Alto	🟠 Alto	Podría redirigir usuarios a sitios falsos.

Robo de sesión (sin HTTPS)	Media	Medio	🟡 Medio	Solo en redes no confiables.
Corte eléctrico reiniciando MySQL	Baja	Medio	🟡 Medio	Puede causar corrupción de tablas.
MySQL expuesto por error a la red	Baja	Alto	🟡 Medio	Riesgo si se cambia bind-address.
Subida de imágenes maliciosas	Baja	Bajo	🟢 Bajo	No afecta a la integridad del sistema.
Ataques al generador Python/PDF	Baja	Bajo	🟢 Bajo	Su impacto es limitado.
Configuración incorrecta del VirtualHost	Media	Bajo	🟡 Medio	Provoca errores de acceso a rentacar.local.

Riesgos más relevantes

1. 🔴 Pérdida de la base de datos
2. 🔴 Inyección SQL
3. 🔴 Acceso no autorizado (HTTP sin HTTPS)
4. 🔴 Exposición de credenciales o archivos sensibles en GitHub
5. 🟠 Errores humanos en la gestión de reservas

Interpretación

- **Los riesgos críticos deben ser tratados de forma inmediata: validación de formularios, backups, cifrado, revisión del control de accesos.**
- **Los riesgos altos deben mitigarse durante el despliegue y operación del sistema.**
- **Los riesgos medios deben monitorizarse.**
- **Los riesgos bajos se aceptan sin acciones adicionales.**

---

5. Plan de Mitigación de Riesgos

El objetivo de este plan es definir medidas preventivas y correctivas para reducir la probabilidad e impacto de los riesgos identificados.

5.1 Medidas para Riesgos Críticos

1. Pérdida total de la base de datos

Medidas:

- **Configurar un sistema de copias automáticas de MySQL:**
    - **Backup diario en local.**
    - **Copia semanal en GitHub privado o memoria USB cifrada.**
- **Exportación periódica mediante `mysqldump`.**
- **Separar base de datos y código en carpetas diferentes.**

---

2. Inyección SQL

Medidas:

- **Usar siempre sentencias preparadas en PHP (`mysqli_prepare` o PDO).**
- **Validar y sanear todos los datos de entrada.**

- **Rechazar caracteres especiales peligrosos.**
- **Deshabilitar mensajes de error SQL visibles en producción.**

---

3. Acceso no autorizado (HTTP sin HTTPS)

Medidas:

- **Instalar un certificado SSL autofirmado en Apache.**
- **Forzar acceso mediante https://rentacar.local.**
- **Activar `session.cookie_secure = true`.**
- **Limitar duración de sesiones.**

---

4. Exposición de credenciales en GitHub

Medidas:

- **Revisar `.gitignore` para no subir archivos sensibles.**
- **Nunca incluir claves SMTP o contraseñas en el repositorio.**
- **Utilizar variables de entorno si fuera necesario.**
- **Configurar repositorio como privado si procede.**

---

5. Errores humanos en la gestión de reservas

Medidas:

- **Añadir confirmación (`confirm()`) para acciones críticas.**
- **Historial de logs de cambios.**
- **Interfaz mejorada para evitar confusiones.**
- **Rol "empleado" con permisos limitados.**

5.2 Medidas para Riesgos Altos

Fallo de disco

- **Copias en unidades externas.**
- **Verificar periódicamente estado SMART del disco.**

Problemas con SMTP

- **Implementar reintentos automáticos.**
- **Validar antes de enviar.**

Modificación del archivo hosts

- **Proteger archivo con permisos elevados.**
- **Documentar ruta y configuración en el TFG.**

5.3 Medidas para Riesgos Medios

Robo de sesión

- **Regenerar ID de sesión en cada login.**
- **Expiración automática de sesión en 20–30 minutos.**
- **Bloqueo de cuenta tras X intentos fallidos.**

Corte eléctrico

- **Guardar logs y copias automáticas al apagar.**
- **Uso de un SAI si se despliega en un entorno real.**

5.4 Medidas para Riesgos Bajos

- **Control de integridad básico sobre imágenes.**
- **Validar tipos MIME al subir fotos.**
- **Permisos correctos en carpetas (`644` / `755`).**

---

6. Plan de Continuidad y Copias de Seguridad

Este apartado garantiza que el sistema puede recuperarse de un error grave o desastre.

6.1 Copias de Seguridad

Base de datos MySQL

- **Backup automático diario:**
    - **Script programado con `mysqldump`.**

Comando:

```
mysqldump -u root gestion_reservas > backup_$(date +%F).sql
```

    -
- **Backup semanal externo:**
    - **USB cifrada o nube privada.**
- **Retención:**
    - **Diarios → 7 días.**
    - **Semanales → 1 mes.**

Código del proyecto

- **Copias en GitHub.**
- **Cada actualización requiere commit + push.**
- **Uso de ramas para cambios mayores.**

Imágenes y archivos

- **Guardadas en `img/` y copiadas en el backup.**

6.2 Procedimiento de recuperación

1. **Instalar XAMPP en un equipo nuevo.**
2. **Restaurar carpetas del proyecto (`gestion_reservas`).**
3. **Importar la última copia SQL desde phpMyAdmin.**
4. **Verificar conexión MySQL + Apache.**
5. **Probar login de clientes y empleados.**
6. **Regenerar certificados SSL si existían.**

6.3 Continuidad de servicio

Aunque el sistema no es crítico al nivel empresarial, se garantiza:

- **Disponibilidad del servicio siempre que el servidor esté encendido.**
- **Reinicio manual de Apache/MySQL ante fallos.**
- **Creación de logs para analizar incidentes.**

- **Monitorización manual por parte del administrador.**

6.4 Escenario de desastre

En caso de pérdida total del equipo:

- **Se reinstala XAMPP en otro PC.**
- **Se clona el repositorio GitHub.**
- **Se importa el backup SQL más reciente.**
- **Se restaura funcionalidad en menos de 1 hora.**

---

7. Plan de Mejora Continua y Seguimiento

Para mantener la seguridad y calidad del sistema, se establece un proceso de mejora continua basado en revisiones periódicas.

7.1 Revisión periódica de seguridad

Elemento	Frecuencia	Acción
Base de datos	**Semanal**	**Verificar integridad, revisar usuarios.**
Formularios PHP	**Mensual**	**Revisar validaciones y sanitización.**

Backups	**Semanal**	**Comprobar que se ejecutan correctamente.**
Configuración Apache	**Trimestral**	**Comprobar HTTPS, VirtualHost.**
Credenciales SMTP	**Trimestral**	**Cambiar contraseñas.**
Repositorio GitHub	**Mensual**	**Comprobar fugas de información.**

7.2 Mejoras futuras

Mejoras técnicas

- **Migrar de XAMPP a un servidor Linux real (Ubuntu Server).**
- **Implementar HTTPS completo con Let's Encrypt (si se expone).**
- **Añadir autenticación de dos factores (2FA).**
- **Crear panel completo de gestión de vehículos desde BD.**

Mejoras funcionales

- **Implementar pasarela de pago real (Stripe, Redsys...).**
- **Añadir API REST para integración con apps móviles.**
- **Añadir logs detallados para auditoría.**

7.3 Seguimiento del sistema

El administrador realizará:

- **Revisión mensual del funcionamiento del sistema.**
- **Pruebas de reserva como cliente y empleado.**
- **Verificación de envío de correos.**
- **Prueba de regeneración de informes PDF.**

Si se detecta un fallo:

1. **Se registra el incidente (hora, afectación, pasos previos).**
2. **Se aplica mitigación.**
3. **Se documenta la solución.**
4. **Se actualiza el Plan de Riesgos si procede.**
