# Técnicas de seguridad de datos

## Cifrado de datos

El cifrado es un método de protección de datos que consiste en alterarlos hasta hacerlos ilegibles. Los datos pasan de ser texto sin formato a ser texto cifrado por medio de un método denominado algoritmo. Quien desee acceder a los datos cifrados debe descodificarlos primero con la clave de descifrado correcta.

El cifrado funciona pasando los datos originales (o texto plano) por un algoritmo (o cifra) que los convierte en texto cifrado. El nuevo texto resulta ilegible si no se utiliza la clave de descifrado adecuada para descodificarlo.

Esto es igualmente cierto para los datos en reposo (almacenados en algún sitio, como un disco duro) y en movimiento (en transferencia electrónica de un lugar a otro, por ejemplo, a través de una red o de Internet).

Los algoritmos de cifrado emplean claves criptográficas (cadenas de caracteres) para transformar los datos en un sinsentido aparentemente aleatorio. Los algoritmos modernos descomponen los datos de texto plano en grupos llamados bloques y luego cifran cada bloque como una unidad. Por este motivo se los denomina cifradores de bloques.

- BCrypt Algoritmo de cifrado

  La contraseña de la tabla de usuario generalmente se cifra mediante un algoritmo irreversible como MD5 y luego se almacena. Para evitar que la tabla de arco iris se rompa, se utiliza una cadena específica (como un nombre de dominio) para el cifrado, y luego se utiliza una sal aleatoria (valor de sal) para el cifrado. La cadena específica se fija en el código del programa, y la sal es aleatoria separada para cada contraseña. Generalmente, es problemático agregar un campo a la tabla de usuario para almacenar por separado. El algoritmo BCrypt aleatoriza la sal y la mezcla en la contraseña cifrada final, y no es necesario proporcionar la sal anterior por separado durante la verificación, por lo que no es necesario tratar el problema de la sal por separado

Cifrar al momento de insertar en la bd

```
# encriptamos la contraseña
contra = contra.encode('utf-8')
hashed = bcrypt.hashpw(contra, bcrypt.gensalt())
```

Como es irreversible se usa checkpw

```
passBD=passBD.encode('utf-8')
if bcrypt.checkpw(password,passBD)
```

## Respaldo y recuperación de bases de datos

En general, copia de seguridad y recuperación se refieren a las diversas estrategias y procedimientos involucrados en la protección de tu base de datos contra la pérdida de información, y la reconstrucción después de cualquier tipo de pérdida.

Una copia de seguridad es una copia tu base de datos que se puede usar para reconstruirla. Se pueden dividir en copias físicas y copias lógicas.

Copias de seguridad físicas de base de datos

Las copias de seguridad físicas son respaldos de los archivos físicos utilizados para almacenar y recuperar tu información, como archivos de datos, archivos de control y registros. En última instancia, cada copia de seguridad física es una copia de archivos que almacenan información de la base de datos en otra ubicación, ya sea en un disco o en algún almacenamiento físico como la cinta.

Copias de seguridad lógicas de base de datos

Las copias de seguridad lógicas contienen datos lógicos (por ejemplo, tablas o procedimientos almacenados) exportados desde una base de datos y almacenados en un archivo binario, para luego volver a importarlos a una base de datos utilizando la utilidad de importación correspondiente.

En la técnica de respaldo lógico, las utilidades importar / exportar se utilizan para crear el respaldo de la base de datos. Una copia de seguridad lógica realiza una copia de seguridad del contenido.

## Procedimiento con el cual se realizó una copia de seguridad de manera local

- El comando que se utilizará es el siguiente: mysqldump -u usuario -p password dbname > C:\respaldo\dbname.sql
- Se deberá tener configurado mysql en el path del sistema para ello debemos ir a Configuración avanzada del sistema>opciones avanzadas>variables de entorno>variables del sistema>path>editar> y seleccionar la carpeta bin de la instalación del mysql.
- Guardar el comando en algún lugar, con un nombre.bat, que es un ejecutable que reconoce Windows
- Se puede jalar a la línea de comandos el archivo y presionar enter.o simplemente darle doble click y se ejecutará de manera automática.
- Para hacer que se ejecute de manera automática deberemos ir al programador de tareas> biblioteca del programador de tareas
- Click en crear tarea básica > Colocar el nombre de la tarea, descripción de la tarea
- Click en siguiente> elegir periodicidad
- Click en siguiente> seleccionar la fecha de inicio, seleccionar la hora especifica en la que se va a ejecutar la tarea
- Click en Acción > Iniciar un programa > siguiente

- Presionar examinar> buscar el .bat que se ha generado de manera previa, no agregar nada en los argumentos.
- Siguiente> finalizar
- Modificar el .bat si se quiere conocer la fecha del respaldo por ejemplo
  set year=%date:~6,4%
  set month=%date:~3,2%
  set day=%date:~0,2%
  set name=name_%year%%month%%day%.sql
  mysqldump -u user -p password dbname > route\%name%

## Enmascaramiento de datos

El enmascaramiento de datos es un método con el que podemos crear una versión que tiene una estructura similar a la de los datos originales pero que no es auténtica y que puede utilizarse para fines tales como pruebas de software y formación de usuarios. El propósito de esto es proteger los datos reales a la vez que se dispone de un sustituto funcional para ocasiones en las que los datos reales no son necesarios.

Aunque la mayoría de organizaciones tienen estrictos controles de seguridad para proteger los datos de producción, tanto en su lugar de almacenamiento como cuando se están utilizando en el negocio, algunas veces los mismos datos son utilizados para operaciones que no son del todo seguras. El problema a menudo se puede complicar si estas operaciones son subcontratadas a empresas externas donde la organización tiene poco control sobre lo que se hace allí con los datos. Para cumplir con los requisitos legales la mayoría de organizaciones no se sienten cómodas exponiendo innecesariamente sus datos reales.

En el enmascaramiento de datos, el formato de los datos sigue siendo el mismo. Sólo se cambian los valores. Los datos pueden ser cambiados de diferentes formas incluyendo la encriptación, la mezcla de caracteres, o la sustitución de palabras. Cualquier método que sea elegido, debe garantizar que los valores son modificados de forma que se imposibilite el descubrimiento del valor real o la posibilidad de hacer ingeniería inversa.

## Análisis de vulnerabilidades

Los hackers suelen analizar las redes de forma activa o pasiva en busca de agujeros y vulnerabilidades. Los analistas de seguridad de datos y los profesionales de la evaluación de vulnerabilidades son elementos clave en la identificación de posibles agujeros y en cerrarlos. El software de análisis de seguridad se utiliza para aprovechar cualquier vulnerabilidad de un ordenador, red o infraestructura de comunicaciones, priorizando y abordando cada uno de ellos con planes de seguridad de datos que protegen, detectan y reaccionan.

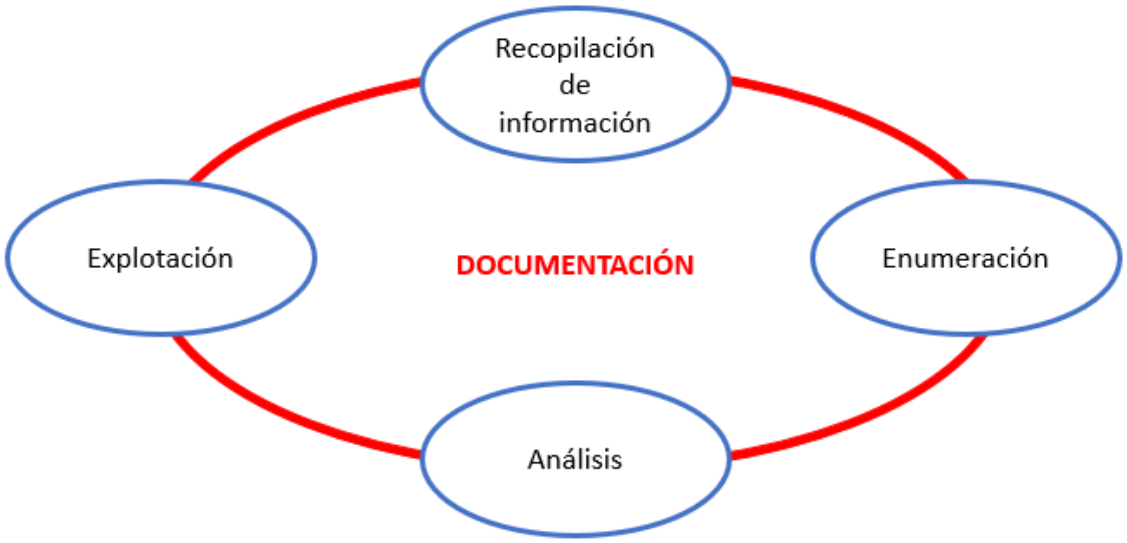
## Pruebas de intrusión

El análisis de vulnerabilidad (que identifica amenazas potenciales) también puede incluir deliberadamente investigar una red o un sistema para detectar fallos o hacer pruebas de intrusión. Es una excelente manera de identificar las vulnerabilidades antes de tiempo y diseñar un plan para solucionarlas. Si hay fallos en los sistemas operativos, problemas con incumplimientos, el código de ciertas aplicaciones u otros problemas similares, un administrador de red experto en pruebas de intrusión puede ayudarte a localizar estos problemas y aplicar parches para que tengas menos probabilidades de tener un ataque.

Las pruebas de intrusión implican la ejecución de procesos manuales o automatizados que interrumpen los servidores, las aplicaciones, las redes e incluso los dispositivos de los usuarios finales para ver si la intrusión es posible y dónde se produjo esa ruptura. A partir de esto, pueden generar un informe para los auditores como prueba de cumplimiento.

Una prueba de intrusión completa puede ahorrarte tiempo y dinero al prevenir ataques costosos en áreas débiles que no conoces. El tiempo de inactividad del sistema puede ser otro efecto secundario molesto de ataques maliciosos, por lo que hacer pruebas de intrusión con regularidad es una excelente manera de evitar problemas antes de que surjan.

Dichas pruebas las haremos de la siguiente manera:




## Información de seguridad y gestión de eventos

Hay una línea aún más holística de defensa que se puede emplear para mantener los ojos en cada punto de contacto. Es lo que se conoce como Información de Seguridad y Gestión de Eventos (SIEM). SIEM es un enfoque integral que

monitoriza y reúne cualquier detalle sobre la actividad relacionada con la seguridad de TI que pueda ocurrir en cualquier lugar de la red, ya sea en servidores, dispositivos de usuario o software de seguridad como NIDS y firewalls. Los sistemas SIEM luego compilan y hacen que esa información esté centralizada y disponible para que se pueda administrar y analizar los registros en tiempo real, e identificar de esta forma los patrones que destacan.

Estos sistemas pueden ser bastante complejos de configurar y mantener, por lo que es importante contratar a un experto administrador SIEM.

### Ciberseguridad: HTTPS, SSL y TLS

Internet en sí mismo se considera una red insegura, lo cual es algo que puede asustar cuando nos damos cuenta que actualmente es la espina dorsal de muchas de las transacciones de información entre organizaciones. Para protegernos de que, sin darnos cuenta, compartamos nuestra información privada en todo Internet, existen diferentes estándares y protocolos de cómo se envía la información a través de esta red. Las conexiones cifradas y las páginas seguras con protocolos HTTPS pueden ocultar y proteger los datos enviados y recibidos en los navegadores. Para crear canales de comunicación seguros, los profesionales de seguridad de Internet pueden implementar protocolos TCP/IP (con medidas de criptografía entretejidas) y métodos de encriptación como Secure Sockets Layer (SSL) o TLS (Transport Layer Security).

El software anti-malware y anti-spyware también es importante. Está diseñado para supervisar el tráfico de Internet entrante o el malware como spyware, adware o virus troyanos.

### Detección de amenazas en punto final

Se pueden prevenir ataques de ransomware siguiendo buenas prácticas de seguridad, como tener software antivirus, el último sistema operativo y copias de seguridad de datos en la nube y en un dispositivo local. Sin embargo, esto es diferente para organizaciones que tienen múltiple personal, sistemas e instalaciones que son susceptibles a ataques.

Los usuarios reales, junto con los dispositivos que usan para acceder a la red (por ejemplo, teléfonos móviles, ordenadores portátiles o sistemas TPV móviles), suelen ser el eslabón más débil de la cadena de seguridad. Se deben implementar varios niveles de protección, como tecnología de autorización que otorga acceso a un dispositivo a la red.

### Prevención de pérdida de datos (DLP)

Dentro de la seguridad de punto final hay otra estrategia de seguridad de datos importante: la prevención de pérdida de datos (DLP). Esencialmente, esto abarca las medidas que se toman para asegurar que no se envían datos confidenciales desde la red, ya sea a propósito, o por accidente. Puede implementarse software

DLP para supervisar la red y asegurarse de que los usuarios finales autorizados no estén copiando o compartiendo información privada o datos que no deberían.

## Técnicas de mantenimiento de software

El mantenimiento del software es el proceso de cambiar, modificar y actualizar el software para mantenerse al día con las necesidades del cliente. El mantenimiento del software se realiza después de que se haya lanzado el producto por varias razones, incluida la mejora del software en general, la corrección de problemas o errores, para mejorar el rendimiento y más.

El mantenimiento del software es una parte natural del SDLC (ciclo de vida de desarrollo de software). Los desarrolladores de software no pueden darse el lujo de lanzar un producto y dejar que se ejecute, necesitan estar constantemente atentos para corregir y mejorar su software para seguir siendo competitivos y relevantes.

El uso de las técnicas y estrategias de mantenimiento de software adecuadas es una parte fundamental para mantener cualquier software en funcionamiento durante un largo período de tiempo y mantener contentos a los clientes y usuarios.

Cada uno de los cuatro tipos diferentes de mantenimiento de software se realiza por diferentes razones y propósitos. Una determinada pieza de software puede tener que someterse a uno, dos o todos los tipos de mantenimiento a lo largo de su vida útil.

Los cuatro tipos son:

- Mantenimiento correctivo de software Mantenimiento
- preventivo de software Mantenimiento
- perfectivo de software Mantenimiento
- adaptativo de software

## Mantenimiento Correctivo de Software

El mantenimiento correctivo de software es la forma típica y clásica de mantenimiento (para software y cualquier otra cosa). El mantenimiento correctivo del software es necesario cuando algo sale mal en una pieza de software, incluidas fallas y errores. Estos pueden tener un impacto generalizado en la funcionalidad del software en general y, por lo tanto, deben abordarse lo más rápido posible.

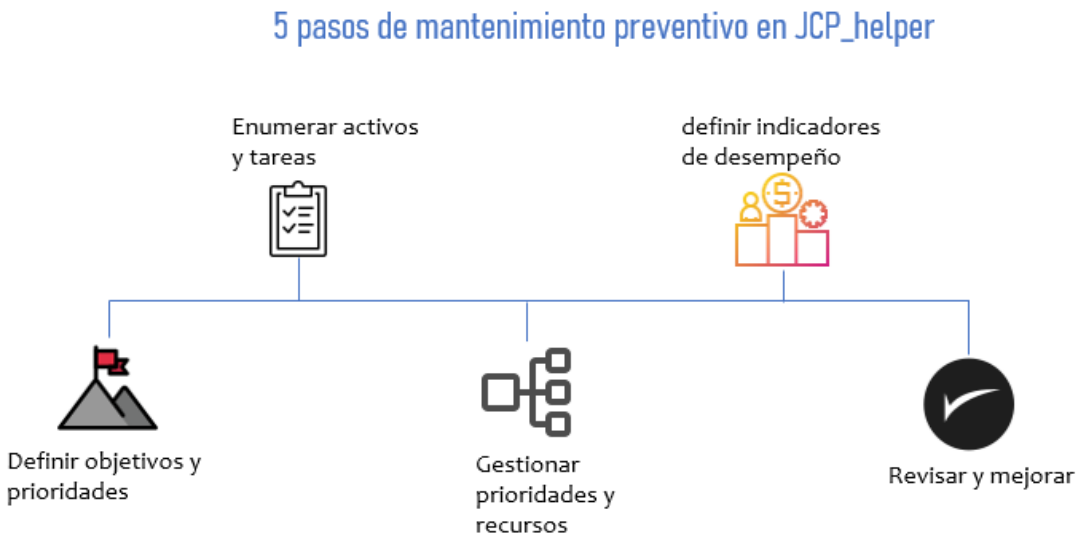
Muchas veces, los proveedores de software pueden abordar problemas que requieren mantenimiento correctivo debido a los informes de errores que envían los usuarios. Si una empresa puede reconocer y solucionar los errores antes de que los usuarios los descubran, esta es una ventaja adicional que hará que su empresa

parezca más respetable y confiable (a nadie le gusta un mensaje de error después de todo).

## Mantenimiento Preventivo de Software

El mantenimiento preventivo de software mira hacia el futuro para que su software pueda seguir funcionando como se desea durante el mayor tiempo posible.

Esto incluye hacer los cambios necesarios, actualizaciones, adaptaciones y más. El mantenimiento preventivo del software puede abordar pequeños problemas que en un momento dado pueden carecer de importancia pero que pueden convertirse en problemas mayores en el futuro. Estos se denominan fallas latentes que deben detectarse y corregirse para asegurarse de que no se conviertan en fallas efectivas.




## Mantenimiento Perfectivo de Software

Al igual que con cualquier producto en el mercado, una vez que el software se lanza al público, surgen nuevos problemas e ideas. Los usuarios pueden ver la necesidad de nuevas funciones o requisitos que les gustaría ver en el software para que sea la mejor herramienta disponible para sus necesidades. Aquí es cuando entra en juego el mantenimiento perfectivo del software.

El mantenimiento perfectivo de software tiene como objetivo ajustar el software agregando nuevas características según sea necesario y eliminando características que son irrelevantes o no efectivas en el software dado. Este proceso mantiene la relevancia del software a medida que cambian el mercado y las necesidades de los usuarios.

## Mantenimiento de software adaptativo

El mantenimiento de software adaptativo tiene que ver con las tecnologías cambiantes, así como con las políticas y reglas relacionadas con su software. Estos incluyen cambios en el sistema operativo, almacenamiento en la nube, hardware, etc. Cuando se realizan estos cambios, su software debe adaptarse para cumplir adecuadamente con los nuevos requisitos y continuar funcionando bien.

La mayoría de los modelos de procesos de mantenimiento de software incluyen los siguientes pasos:

1. Identificación y seguimiento: el proceso de determinar qué parte del software debe modificarse (o mantenerse). Esto puede ser generado por el usuario o identificado por el propio desarrollador de software según la situación y la falla específica.
2. Análisis: el proceso de analizar la modificación sugerida, incluida la comprensión de los efectos potenciales de dicho cambio. Este paso generalmente incluye un análisis de costos para comprender si el cambio vale la pena desde el punto de vista financiero.
3. Diseño: diseñar los nuevos cambios utilizando especificaciones de requisitos.
4. Implementación: el proceso de implementación de los nuevos módulos por parte de los programadores.
5. Pruebas del sistema: antes de iniciarse, se deben probar el software y el sistema. Esto incluye el módulo en sí, el sistema y el módulo, y todo el sistema a la vez.
6. Pruebas de aceptación: los usuarios prueban la modificación para su aceptación. Este es un paso importante ya que los usuarios pueden identificar problemas en curso y generar recomendaciones para una implementación y cambios más efectivos.
7. Entrega: actualizaciones de software o, en algunos casos, nueva instalación del software. Aquí es cuando los cambios llegan a los clientes.

## Costo de mantenimiento de software

El costo del mantenimiento del software puede ser alto. Sin embargo, esto no niega la importancia del mantenimiento del software. En ciertos casos, el mantenimiento del software puede costar hasta dos tercios del ciclo completo del proceso de software o más del 50 % de los procesos SDLC.

Los costos involucrados en el mantenimiento del software se deben a múltiples factores y varían según la situación específica. Cuanto más antiguo sea el software, más costará el mantenimiento, ya que las tecnologías (y los lenguajes de codificación) cambian con el tiempo. Renovar una pieza de software antigua para adaptarse a la tecnología actual puede ser un proceso excepcionalmente costoso en ciertas situaciones.

Además, es posible que los ingenieros no siempre puedan abordar los problemas exactos cuando buscan actualizar o mantener una pieza específica de software. Esto hace que utilicen un método de prueba y error, que puede resultar en muchas horas de trabajo.

Existen ciertas formas de intentar reducir los costos de mantenimiento del software. Estos incluyen la optimización de la parte superior de la programación utilizada en el software, la tipificación fuerte y la programación funcional.

Al crear un nuevo software, así como al asumir proyectos de mantenimiento para modelos más antiguos, las empresas de software deben tener en cuenta los costos de mantenimiento del software. Sin mantenimiento, cualquier software quedará obsoleto y esencialmente inútil con el tiempo.

Estrategias de mantenimiento de software

Todas las empresas de software deben tener una estrategia específica para abordar el mantenimiento del software de manera efectiva y completa.

La documentación es una estrategia importante en el desarrollo de software. Si la documentación del software no está actualizada, la actualización puede parecer imposible. La documentación debe incluir información sobre cómo funciona el código, soluciones a problemas potenciales, etc.

El control de calidad también es una parte importante de un plan de mantenimiento de software. Si bien el control de calidad es importante antes del lanzamiento inicial de un software, también se puede integrar mucho antes en el proceso (ya en la etapa de planificación) para garantizar que el software se desarrolle correctamente y brindar información sobre cómo realizar cambios cuando sea necesario.

# PROCEDIMIENTO MANTENIMIENTO CORRECTIVO Y PREVENTIVO DE JCP_HELPER

## 1.- OBJETIVO

Garantizar el correcto funcionamiento de JCP_helper en diferentes dispositivos y con los diferentes tipos de usuarios mediante la realización del mantenimiento de software con asistencia técnica preventiva y correctiva.

## 2.-ALCANCE

El siguiente procedimiento aplicara a todos los módulos y submodulos de la aplicación, así como a la base de datos de JCP_helper.

## 3.- DEFINICIONES

3.1 SOFTWARE: equipamiento o soporte lógicos a todos los componentes intangibles de un computador, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

3.2. MANTENIMIENTO: Conservación de una cosa en buen estado o en una situación determinada para evitar su degradación.

3.3. MANTENIMIENTO CORRECTIVO: Es aquel mantenimiento que se realiza con el fin de corregir o reparar un fallo en el equipo o instalación.

3.4. MANTENIMIENTO PREVENTIVO: Es aquel que se realiza de manera anticipado con el fin de prevenir el surgimiento de averías en los artefactos, equipos electrónicos, vehículos automotores, maquinarias pesadas, etc.

## 4. GENERALIDADES

4.1 El mantenimiento correctivo se realiza en un máximo de 7 días hábiles después de haber recibido la solicitud, siempre y cuando la solución se encuentre disponible dentro de la app y no sea un problema externo, se le dará prioridad dependiendo de los niveles de servicio.

| NIVELES DE SERVICIOS | |
|---|---|
| Alto | • Problemas con acceso a la aplicación con las credenciales correctas, en ambos tipos de usuarios.<br>• Problemas con la creación de cuestionarios.<br>• Problemas del usuario estudiante, para responder los cuestionarios.<br>• Problemas con la visualización gráfica de la información.<br>• Problemas con el funcionamiento de los algoritmos de machine learning. |
| Medio | • Problemas con la creación y visualización de notificaciones.<br>• Problemas con la recuperación de cuestionarios de la comunidad.<br>• Problemas con la selección de un idioma diferente al español. |
| Bajo | • Problemas con la selección de diferentes temas de la app. |

4.2 El personal de apoyo para el mantenimiento de JCP_helper enviará una solicitud de recursos requeridos en caso de necesitarse para el servicio de mantenimiento, el área Administrativa y financiera.

4.3 El personal debe registrar toda solicitud de mantenimiento correctivo, solicita por el usuario, independientemente sí se puede o no dar solución.

4.4 El responsable de atender la solicitud debe identificar y notificar al usuario el tiempo probable de solución a la solicitud en caso de que no pueda ser atendida al momento.

4.5 Se realizará un reporte de las correcciones a la aplicación, o con las preguntas mas frecuentes para ahorrar esfuerzo a futuro.

## 5. DESCRIPCIÓN NARRATIVA DE LAS ACTIVIDADES.

### 5.1 Mantenimiento correctivo.

| N° | Actividad | Responsable | Documento | Descripción de la actividad |
|---|---|---|---|---|
| 1 | Solicitar el servicio | Usuario solicitante | Correo electrónico | Realizar solicitud de mantenimiento o corrección de errores, mediante el correo electrónico de la app o el número telefónico. |
| 2 | Asignar el servicio | Equipo JCP_helper | NA | El equipo de JCP_helper recibe la solicitud de servicio y se asigna un encargado que se encarga de dar solucionar el problema, dependiendo de la dificultad y requerimientos del problema se asignará mas de un responsable. |

| | | | | |
|---|---|---|---|---|
| 3 | La solución es inmediata | Equipo JCP_helper | Reporte de mantenimiento correctivo. | El equipo soluciona el problema y da respuesta inmediata al usuario, además agrega el problema y la solución a un reporte |
| 4 | Realizar mantenimiento | Equipo JCP_helper | Solicitud de soporte | a) Realizar las actividades de modificación, borrado, adición o actualización, según lo amerite la falla.<br>b) Ejecutar las pruebas técnicas de los cambios realizados al software, verificando que estos no presenten fallas y abarcando lo solicitado.<br>Una vez se dé solución a la falla, el solicitante recibe la notificación sobre la satisfacción del problema recibido, el cual debe ser archivado en el reporte. |

### 5.2 Mantenimiento preventivo.

| N° | Actividad | Responsable | Documento | Descripción de la actividad |
|---|---|---|---|---|
| 1 | Revisión de la app. | Equipo JCP_helper | Reporte mantenimiento preventivo | El equipo de JCP_helper a menudo revisa los diferentes módulos de la aplicación verificando que sigan funcionando adecuadamente |
| 2 | Respaldos de la BD | Equipo JCP_helper | Historial de respaldos | El equipo de JCP_helper a menudo realiza un respaldo de la base de datos de manera física, con el fin de garantizar la integridad y persistencia de los datos. |
| 3 | Cambios de contraseñas | Equipo JCP_helper | Guardar en un lugar seguro | Actualizar credenciales de manera periódica, para que sea difícil para los ciberdelincuentes irrumpir en la parte administrativa de la aplicación. |

## PLAN DE SEGURIDAD

| Objetivos | | Medidas de seguridad | Responsabilidades | Costos | Calendario |
|---|---|---|---|---|---|
| Globales (relacionados con las amenazas) | Específicos (relacionados con las vulnerabilidades | | | | |

| | | | | | |
|---|---|---|---|---|---|
| Reducir las posibilidades de que alguien entre a la parte administrativa de nuestro | Incluso si entran tenemos que evitar que:<br>Se pierda información almacenada en nuestra base de datos<br>Que las personas no autorizadas tengan acceso a información sensible | Clasificar que información es delicada para tomar pasos adicionales para protegerlos de una intrusión | Las personas a cargo de programas y dirección | $0 | En un plazo de 2 semanas |
| | | Elegir un hosting que ofrezca confiabilidad y seguridad en el acceso a datos. | Persona a cargo de TI/ consultor externo de TI | $0 | En un plazo de 1 semana |
| | | Comprar un disco duro externo o un medio de almacenamiento externo | Encargado de finanzas | Variable | En un plazo de una semana |
| Tener un respaldo en físico de la base de datos | Sí por alguna falla técnica en donde estaban alojados los respaldos de la BD, tener uno guardado | Hacer copias/respaldos del disco duro del servidor central | Persona a cargo de información/comunicaciones | $0 | Cada mes |

| | | | | | |
|---|---|---|---|---|---|
| | en un lugar seguro. | | | | |
| Tener seguridad en la aplicación, mediante la encriptación de información. | Dificultar a los posibles atacantes ingresar a la parte administrativa de la aplicación. Aun si logran entrar mostrar la información sensible de nuestros usuarios encriptada y no en texto plano. Evitar comprometer información sensible. | Identificar un método o programa de encriptación. | Persona a cargo de información/comunicaciones | $0 | En un plazo de un 15 días |
| El equipo JCP, esta al tanto de la importancia de la encriptación. | Ser capaces de aplicar la encriptación en la aplicación. Generar conciencia para el uso de contraseñas robustas difíciles de descifrar. | Capacitación interna sobre el programa de encriptación y contraseñas robustas. | Todos los miembros del esquipo jcp_helper | $0 | En un plazo de 15 días |
| Evitar la irrupción en el servidor central y | Evitar el acceso de personas externas a JCP al | Tomar medidas de seguridad en el servidor | Persona a cargo de información/comunicaciones | $0 | En un plazo de dos semanas. |

| | | | | | |
|---|---|---|---|---|---|
| en los respaldos. | servidor central. Evitar que los respaldos sean accedidos por personas no autorizadas. Preservar la integridad y confidencialidad de información. | central y en los respaldos. | | | |
| | | Denunciar la irrupción al sistema de personas externas al equipo JCP_helper. | Persona a cargo de información/comunicaciones | $0 | Una semana posterior a la intrusión. |
| Encontrar a los responsables de las irrupciones en el sistema. | Tomar las medidas legales o sanciones correspondientes en contra de los responsables. Si se presentaron daños, hacer que los responsables hagan la reposición de los daños con | Hacer presión a las autoridades correspondientes para investigar a los dispositivos que están detrás de las irrupciones, para determinar quienes son los responsables y que | Persona a cargo de las incidencia | $0 | De inmediato |

| | | | | | |
|---|---|---|---|---|---|
| | aquellos que fueron perjudicados. | sean inculpados. | | | |
| Encriptar la comunicación entre el servidor y el navegador web | Hacer que la aplicación cumpla con los últimos estándares de seguridad.<br>Hacer una aplicación confiable para los usuarios.<br>Evitar que los piratas informáticos intercepten información delicada en el navegador. | Utilizar el protocolo HTTPS, y un certificado digital. | Persona a cargo de TI | Variable | 1 semana |
| Verificar y validar la funcionalidad de los módulos de la aplicación | Hacer un proceso de verificación de funcionalidad de los módulos que han presentado fallos.<br>Hacer proceso de verificación de seguridad de los módulos restantes para evitar | Hacer testing en los módulos de la aplicación. | Encargado de TI | $0 | Periódicamente |

| | | | | | |
|---|---|---|---|---|---|
| | fallos futuros | | | | |