| **ITEM** | **Cumple** | **No cumple** | **No aplica** | **Alcance** | **Impacto** |
| --- | --- | --- | --- | --- | --- |
| Inyección SQL | X |  |  | Se realizo inyección SQL en los elementos de entrada tales como los inputs | El sistema valido correctamente los datos ingresados en los elementos de entrada mostrando error al usuario de ser necesario |
| Pérdida de autenticación | X |  |  | Se tuvieron en cuenta los elementos con los cuales debía contar el sistema, como el borrar cache al cerrar sesión, además de contar con la posibilidad de cerrar sesión. | El sistema cumplia con las recomendaciones dadas por el OWASP. |
| Exposisicón de datos sensibles | X |  |  | Se evaluaron los datos expuestos en el navegador, mediante el uso de la consola del mismo, para poder capturar datos sensibles del sistema. | El sistema cumplía con las recomendaciones del documento, al mantener toda la información sensible del sistema confidencial. |
| Entidades externas XML |  |  | X |  |  |
| Pérdida de control de acceso | X |  |  | Se encriptan todos los datos mediante tokens en el backend y bloqueando el acceso al sistema de usuarios no autenticados o permitidos, la aplicación al no hacer uso de api o servicios externos no se realizaron mas pruebas. | El sistema impidió correctamente el acceso a usuario no autorizados. |
| Configuración de seguridad incorrecta | X |  |  | Se manejaron los errores generados por el sistema, para mostrarle al usuario el error, se desactivaron cuentas las cuales se crearon para el uso en desarrollo y para realizar pruebas, se desactivaron funciones y módulos los cuales ya no se estaban usando. | El sistema muestra mensajes de error al usuario indicando su falla sin comprometer información del sistema, además el sistema elimino todas las cuentas usadas en desarrollo y desactivo satisfactoriamente las funcionalidades del sistema las cuales no serian mas usadas. |
| Secuencia de comandos en sitios cruzados | X |  |  | Se intento ingresar datos y fragmentos de código en los formulario para ver como respondía el sistema, si ejecuta código malicioso o validaba primero los datos ingresados por el usuario. | El sistema informaba al usuario cuando encontraba datos ingresados erróneamente, evitando así los fragmentos de código maliciosos. |
| Deserialización insegura | X |  |  | Se intento incrustar código en formato de Json, ya que el sistema a pesar de no hacer uso de apis o servicios externos, si realiza envíos y capturas de datos en formato json. | El sistema valido estos datos al ser ingresados, lo cual no dejo manipular la lógica del sistema, ya que la transferencia de datos en formato de tipo Json se hace usando un token, lo cual impide este tipo de ataques. |
| Componentes con vulnerabilidades conocidas | X |  |  | Se realizo una búsqueda en el sistema en busca de librerías con vulnerabilidades conocidas. | Se evidencio que el uso de Jquery generaba una vulnerabilidad conocida pero en una versión anterior a la cual se estaba usando en el sistema, ya que fue corregida en su ultima versión. |
| Registro y Monitoreo Insuficientes | X |  |  | Se realizaron pruebas las cuales buscaban generar errores o actividad en el sistema. | El sistema almacena toda está actividad en un documento interno ‘log’ en cual se puede visualizar la trasabilidad del sistema |