

CLIENTE: Perenco

Revisión Técnica del "Sistema Integral"

09 de Febrero 2010

AUREAL



Tabla de contenidos

Análisis técnico del sistema actual	6
Revisión de Base de datos	6
Nombramiento inconsistente de tablas y columnas	6
Situación encontrada	6
Riesgo	6
Recomendación	6
Gravedad	6
Motor sin soporte de transacciones	7
Situación	7
Riesgo	7
Recomendación	7
Gravedad	7
Uso de Charset con problemas de compatibilidad	7
Situación	7
Riesgo	7
Recomendación	7
Gravedad	7
Tablas en desuso	8
Situación	8
Riesgo	8
Recomendación	8
Gravedad	8
Uso de llaves ineficientes	8
Situación	8
Riesgo	8
Recomendación	8
Gravedad	8
Revisión de Seguridad	9
Mal uso de funciones de control de acceso	9
Situación	9
Riesgo	9
Recomendación	9
Gravedad	9
Problemas de inyección SQL	9
Situación	9
Riesgo	9
Recomendación	9
Gravedad	10
No se cuenta con protección adecuada para las funcionalidades	10
Situación	10
Riesgo	10
Recomendación	10
Gravedad	10
Revisión de eficiencia de carga	10
Lentitud en los reportes	10
Situación	10
Riesgo	10

Recomendación	10
Gravedad	11
Uso de SQL ineficiente	11
Riesgo	11
Recomendación	11
Gravedad	11
Uso de funciones específicas el motor de base de datos	11
Situación	11
Riesgo	11
Recomendación	12
Gravedad	12
Uso excesivo de consultas en reportes	12
Situación	12
Riesgo	12
Recomendación	12
Gravedad	12
Revisión de usabilidad	12
Inconsistencia en las notificaciones de error	12
Situación	12
Riesgo	13
Recomendación	13
Gravedad	13
Los usuarios no están al tanto de la opción de cambiar contraseña	13
Situación	13
Riesgo	13
Recomendación	13
Gravedad	13
Pantallas confusas y sobrecargadas	13
Situación	13
Riesgo	14
Recomendación	14
Gravedad	14
Estructuración desordenada de las funcionalidades	14
Situación	14
Riesgo	14
Recomendación	14
Gravedad	14
No existe información de ayuda para el uso de las funcionalidades	14
Situación	14
Riesgo	15
Recomendación	15
Gravedad	15
Controles mal ubicados	15
Situación	15
Riesgo	15
Recomendación	15
Gravedad	15
No existen mantenimientos	15
Situación	15
Riesgo	16
Recomendación	16

Gravedad	16
Problemas de codificación	16
No se ha comentado el código	16
Situación	16
Riesgo	16
Recomendación	16
Gravedad	16
Código redundante	16
Situación	16
Riesgo	17
Recomendación	17
Gravedad	17
Mala indentación del código	17
Situación	17
Riesgo	17
Recomendación	17
Gravedad	17
No hay separación de capas de lógica	17
Situación	17
Riesgo	17
Recomendación	18
Gravedad	18
Nomenclatura inconsistente	18
Situación	18
Riesgo	18
Recomendación	18
Gravedad.....	18
Mezcla de lenguajes utilizados en un mismo archivo de presentación.....	18
Situación	18
Riesgo	18
Recomendación	19
Gravedad.....	19
Problemas con Javascript	19
Uso de sisntaxis inconsistente	19
Situación	19
Riesgo	19
Recomendación	19
Gravedad	19
Uso de métodos no estándar	19
Situación	19
Riesgo	19
Recomendación	20
Gravedad	20
Inclusión y definición de funciones múltiples	20
Situación	20
Riesgo	20
Recomendación	20
Gravedad	20
Archivos mezclados en directorios	20
Situación	20
Riesgo	20

Recomendación	21
Gravedad	21
HTML	21
Documentos HTML declarados sin Doctype	21
Situación	21
Riesgo	21
Recomendación	21
Gravedad	21
No se ha definido un encoding para los documentos	21
Situación	21
Riesgo	21
Recomendación	21
Gravedad	22
Separación incorrecta de presentación de datos	22
Riesgo	22
Recomendación	22
Gravedad	22
HTML no válido	22
Riesgo	22
Recomendación	22
Gravedad	22
CSS	23
Referencias a archivos inexistentes	23
Riesgo	23
Recomendación	23
Gravedad	23
CSS inválido	23
Riesgo	23
Recomendación	23
Gravedad	23

Análisis técnico del sistema actual

En la videoconferencia llevada a cabo el día 2 de Febrero con los equipos de Perenco Lima y Perenco Francia, se nos solicitó realizar la revisión del "Sistema Integral" que actualmente se encuentra en funcionamiento.

Para la revisión se tomaron en cuenta los siguientes aspectos:

- Base de datos
- Seguridad
- Eficiencia
- Usabilidad
- Codificación
 - Javascript
 - HTML
 - CSS

Revisión de Base de datos

Nombramiento inconsistente de tablas y columnas

Situación encontrada

- El nombramiento de tablas no sigue ningún patrón, en algunos casos se utiliza nombres en plural de entidades y en otros, nombres en singular.
- A veces se utiliza el sub-guión '_' como conector entre palabras y otras veces no.
- El nombramiento de columnas tampoco sigue algún patrón, algunas veces se incluye como sufijo la primera parte del nombre de la tabla. Como las columnas de la tabla familiares que terminan en _f. Sin embargo, en otros casos no utilizan ningún prefijo ni sufijo. Algunas veces el código se llama código, otras cod, otras cod_f.

Riesgo

El incorrecto nombramiento de tablas en el sistema dificulta la rápida ubicación de la información al intentar trabajar directamente con cambios en la base de datos

Recomendación

Se recomienda la definición de un estándar para la nomenclatura de tablas y columnas dentro del sistema

Gravedad

Media

Motor sin soporte de transacciones

Situación

Se está utilizando el motor MyISAM de Mysql, el cual no soporta transacciones del tipo ACID y no asegura integridad referencial en los datos.

Riesgo

El no tener soporte de integridad referencial en la base de datos permitiría que se registren datos inconsistentes en la aplicación en caso de cortes abruptos o en caso de mantenimiento incorrecto de los datos, como podría ser la eliminación de un vehículo o persona del sistema quedando huérfanos todos los registros asociados.

Esto generaría errores y el no funcionamiento del sistema.

Recomendación

Se recomienda realizar una evaluación de viabilidad sobre el impacto de migrar o convertir la tabla a usar un motor InnoDB o algún gestor que soporte transaccionalidad.

Gravedad

Baja

Uso de Charset con problemas de compatibilidad

Situación

El charset es latin-1 lo cual puede ocasionar problemas de compatibilidad de caracteres.

Riesgo

El sistema de codificación Latin-1 o ISO8859-1 tiene soporte limitado para caracteres extraños, lo que podría resultar en errores de presentación al momento de imprimir reportes o mostrar datos en pantalla.

Recomendación

Se recomienda modificar el encoding de la base de datos a un esquema UTF-8 para evitar problemas con la información que se pueda ingresar en el sistema.

Gravedad

Baja

Tablas en desuso

Situación

Existe una tabla "temp" parece no tener mucha relación el modelo de datos.

Riesgo

El tener tablas en desuso genera confusión y entorpece los trabajos de depuración y mantenimiento de la base de datos, haciendo que los mantenimientos se vuelvan mas riesgosos y prolongados cada vez.

Recomendación

Se recomienda mantener un diccionario de tablas que permita la rápida identificación de las tablas necesarias por el sistema y permitir el fácil borrado y mantenimiento de ellas.

Gravedad

Baja

Uso de llaves ineficientes

Situación

Algunas de las llaves primarias utilizadas son del tipo VARCHAR haciendo que las consultas y JOINS sean mucho más ineficientes que utilizando valores enteros tipo INTEGER.

Riesgo

El usar datos de tipo VARCHAR para llaves primarias impacta considerablemente a los tiempos de respuesta de la base de datos al momento de realizar consultas en el sistema haciendo que el sistema sea más lento.

Recomendación

Se recomienda utilizar llaves primarias del tipo INTEGER para las referencias entre tablas y así ayudar al motor de base de datos en las consultas realizadas.

Gravedad

Alta

Revisión de Seguridad

Mal uso de funciones de control de acceso

Situación

Las funciones de control de acceso estan replicadas en cada modulo del sistema actual, haciendo que sea muy dificil modificar o cambiar el tipo de control de acceso que se utiliza.

Riesgo

El no contar con un lugar unificado de gestión de seguridad y control de acceso dificulta las labores del administrador del sistema y de mantenimiento para el desarrollado haciendo que cada cambio requerido deba realizarse múltiples veces por cada módulo.

Afecta considerablemente los tiempos de desarrollo y respuesta a cambios en el sistema.

Recomendación

Debería existir un único módulo de gestión de seguridad y control de acceso centralizado para todos los módulos del sistema.

Gravedad

Alta

Problemas de inyección SQL

Situación

No se esta haciendo ningún tipo de limpieza sobre los parámetros recibidos por los formularios GET y POST. Los atacantes podrian enviar sentencias SQL en el formulario, de esta manera se pueden realizar ataques del tipo "SQL injection".

Riesgo

Cualquier usuario con mediano conocimiento de manejo de base de datos podría enviar comandos de corrupción de datos o escalar accesos realizando modificaciones a través del sistema.

Esto permite que un usuario pueda borrar toda la base de datos o realizar modificaciones no autorizadas lo cual afecta la generación de reportes.

Recomendación

Todo ingreso de datos del usuario debería ser validado por el sistema antes de ser enviado a la base de datos para así asegurarse que no se ejecuten comandos maliciosos en el servidor.

Gravedad

Alta

No se cuenta con protección adecuada para las funcionalidades

Situación

Las URLs no están protegidas. Es decir, los permisos sólo se implementan escondiendo o mostrando más opciones en el menú. Sin embargo, si el usuario conoce la URL que desea visitar, puede ingresar incluso sin estar autenticado en el sistema.

Riesgo

El no contar con protección de las funcionalidades del sistema permite que cualquier usuario podría acceder a ellas tan sólo con conocer la dirección. Así pudiendo realizar cambios no autorizados en el sistema.

Recomendación

Debería existir validación de acceso ante la respuesta de cada pantalla más allá de sólo ocultar los links mediante el menú.

Gravedad

Alta

Revisión de eficiencia de carga

Lentitud en los reportes

Situación

Algunos reportes son bastante lentos debido al pobre diseño de la base de datos indicado en la sección "Problemas con la base de datos".

Riesgo

El tener un tiempo de respuesta prolongado genera incomodidad en los usuarios haciendo que el uso del sistema se reduzca y se obtenga información menos precisa con menos detalle.

Recomendación

Se recomienda hacer un análisis de tiempos de respuesta del sistema para identificar los cuellos de botella y poder dirigir esfuerzos específicos en su mejora. Así incrementará el uso del sistema y mejorará la calidad de los datos registrados.

Gravedad

Media

Uso de SQL ineficiente

Situación

- Las sentencias JOIN programadas en las consultas sobretodo de reportes no se hacen mediante JOIN explícitos sino utilizando el producto de dos tablas (Select * from tabla1, tabla2 ...) lo que no permite que la base de datos optimize la consulta.
- Se tienen programadas consultas que buscan y utilizan funciones en SQL para comprar (... where contact(campo1, campo2) like ...) . Estas podrían programarse de una mejor manera para evitar que sean tan lentas.
- Existen consultas programadas hasta con 3 niveles de subconsultas. Probablemente esto se pueda hacer de una forma mas eficiente sin tantas subconsultas.

Riesgo

El hacer mal uso de las sentencias SQL impide al motor de base de datos realizar optimizaciones y fuerza a realizar búsquedas más lentas lo cual afecta considerablemente al tiempo de respuesta del sistema.

Recomendación

Se recomienda hacer un análisis de sentencias utilizadas y reestructurarlas o evaluar el uso de alguna herramienta ORM que realice abstracciones de SQL.

Gravedad

Media

Uso de funciones específicas el motor de base de datos

Situación

Muchos datos se formatean en la base de datos utilizando funciones de Mysql como "contact", "case". Lo que produce un tiempo de respuesta un poco mas lento.

Riesgo

El utilizar comandos específicos de algún motor de base de datos dificulta la portabilidad del sistema a algún otro motor así como tener un pequeño impacto en los tiempos de respuesta al hacer mal uso de dichos comandos que pueden no estar optimizados para consultas muy pesadas.

Recomendación

Se recomienda hacer un análisis de sentencias utilizadas y reestructurarlas o evaluar el uso de alguna herramienta ORM que realice abstracciones de SQL.

Gravedad

Baja

Uso excesivo de consultas en reportes

Situación

Los reportes hacen demasiadas consultas a la base de datos. En casos donde el reporte es por días, se realizan consultas separadas por cada día. Cuando el rango de fechas es considerable, el número de consultas a la base de datos sera enorme.

Riesgo

El generar consultas por cada campo de los reportes implica abrir y realizar lectura de datos por cada valor que se deba calcular lo cual tendrá un alto impacto sobre el tiempo de generación y recursos computacionales del sistema lo cual genera un tiempo de respuesta prolongado para la persona que realiza la consulta y también afectando a los demás usuarios que se encuentren conectados en ese momento.

Recomendación

Se recomienda hacer una reevaluación de las consultas realizadas en SQL para los reportes de tal manera que se realice la menor cantidad de hits a la base de datos para mostrar cada reporte.

Gravedad

Alta

Revisión de usabilidad

Inconsistencia en las notificaciones de error

Situación

El login fallido no muestra ningún mensaje de error.

Muchos errores de validación son mostrados mediante alerts de javascript, deberian indicarse en html para no interferir en la presentación de la pantalla que está usando el usuario.

Riesgo

El no tener una manera uniforme de mostrar los errores en el sistema genera confusión en los usuarios al no saber distinguir entre un tipo de respuesta y otro pudiendo derivar en uso incorrecto de las pantallas del sistema.

Recomendación

Definir una forma estándar de mostrar los errores de ingreso de datos de tal forma que se muestren de una única manera a lo largo del sistema.

Gravedad

Baja

Los usuarios no están al tanto de la opción de cambiar contraseña

Situación

Los usuarios no están al tanto de la posibilidad de cambiar su contraseña, lo que hace que los usuarios se queden con las claves default y al hacerse conocidas pueden ser accedidas por cualquier persona.

Riesgo

Al no cambiar las contraseñas, es fácil de conocer las contraseñas de otros usuarios y así ganar acceso al sistema y realizar cambios no autorizados.

Recomendación

Realizar una capacitación de seguridad a los usuarios explicando la importancia de el correcto mantenimiento de contraseñas en el sistema. Así como también la implementación de generación de contraseñas aleatorias u obligar que los usuarios las cambien al primer ingreso.

Gravedad

Media

Pantallas confusas y sobrecargadas

Situación

Algunos procesos como el de crear/editar manifiestos deberían ser reevaluados por no ser muy fáciles de utilizar y tener demasiados campos en pantalla sirviendo así para muchas cosas a la vez.

Riesgo

El que las pantallas sean reutilizadas para mas de una actividad implica que deban manejar un gran nivel de complejidad lo cual dificulta de gran manera el mantenimiento y tiempo de respuesta a nuevos requerimientos.

Además de ser fuente de mayor cantidad de errores de codificación.

Recomendación

Se recomienda hacer un análisis de las pantallas que se encuentren demasiado cargadas y evaluar la posibilidad de separarlas en pantallas por cada necesidad específica.

Gravedad

Media

Estructuración desordenada de las funcionalidades

Situación

El menú no esta bien estructurado, permitiendo que las funcionalidades se encuentren dispersas sin mayor criterio lógico. (Archivo, Reportes, Seguridad)

Riesgo

El no contar con un menú ordenado o adecuadamente estandarizado permite que futuras funcionalidades desarrolladas sean incorrectamente ubicadas además de impactar sobre la facilidad de uso del sistema haciendo dificultando así la introducción d nuevo personal como usuarios del sistema.

Recomendación

Se recomienda hacer un ordenamiento de las funcionalidades de una manera coherente.

Gravedad

Baja

No existe información de ayuda para el uso de las funcionalidades

Situación

Algunos formularios, como el de bitácora fluvial y mantenimiento de viajes, necesitan un párrafo o texto adicional de ayuda por ser procesos complejos. La información actual es muy pobre.

Riesgo

El no contar con ayuda en pantalla para los usuarios incrementa el riesgo de mal uso de las pantallas generando así información irreal en los reportes generados.

Recomendación

Se recomienda incrementar la cantidad de información sobre el uso de pantallas complejas en el sistema.

Gravedad

Baja

Controles mal ubicados

Situación

Existen acciones mal ubicadas, por ejemplo el caso de "Eliminar" y "Editar" al estar muy juntas harían que algo se pueda eliminar por accidente.

Algunos formularios son demasiado anchos para sus contenedores produciendo que algunos controles se pierdan.

Riesgo

La mala ubicación de acciones podrá generar a la ejecución de comandos no deseados durante la operación del sistema. Lo cual incrementaría el tiempo de retipeo para volver a ingresar los datos.

Recomendación

Se recomienda hacer una evaluación de las posición de controles en pantalla para evitar la confusión de botones.

Gravedad

Media

No existen mantenimientos

Situación

No se cuenta con mantenimientos a varias tablas del sistema por lo que hay que modificar información directamente en la base de datos solicitandole al desarrollador del sistema.

Riesgo

El desarrollador no debería tener acceso a la base de datos de producción del sistema, ya que hacer cualquier cambio directamente puede afectar la integridad de los datos de negocio de la empresa.

En caso no se cuente con la disponibilidad del desarrollador los usuarios se verán imposibilitados de continuar su trabajo de manera actualizada.

Recomendación

Desarrollar módulos de mantenimiento de las entidades principales del sistema

Gravedad

Alta

Problemas de codificación

No se ha comentado el código

Situación

En ninguna parte del código fuente se puede encontrar documentación de código

Riesgo

El no contar con documentación de código dificulta las labores de mantenimiento del sistema así como también genera dependencia de la memoria del programador para conocer el detalle del funcionamiento.

Recomendación

Se recomienda exigir estándares de programación que incluyan la oportuna documentación del código escrito.

Gravedad

Media

Código redundante

Situación

- Se ha encontrado gran cantidad de redundancia de código
- Existen archivos que parecen ser duplicados

Riesgo

El contar con código redundante incrementa la complejidad y tiempo de mantenimiento del sistema así como ser fuente de inconsistencia en el manejo de información al existir bugs en distintas copias de un mismo código.

Recomendación

Se recomienda hacer un mapeo de código del sistema y un trabajo de refactorización de código para asegurarse que no se esté duplicando la funcionalidad.

Gravedad

Alta

Mala indentación del código

Situación

El código no se encuentra claramente indentado.

Riesgo

La mala indentación de código afecta la legibilidad del mismo dificultando las labores de mantenimiento y reacción ante nuevos requerimientos del negocio.

Recomendación

Someter el código a herramientas de revisión de estilo de programación para realizar una correcta indentación.

Gravedad

Baja

No hay separación de capas de lógica

Situación

No existe una separación coherente de las distintas capas del programa

Riesgo

El no contar con una separación por capas conlleva a escribir código de manera desordenada dificultando así la separación de partes del proyecto entre más de un programador. Impidiendo la ampliación del equipo y retardando tiempos de reacción ante modificaciones del sistema.

Recomendación

Evaluar la posibilidad de comenzar a utilizar algún framework del lenguaje que permita hacer una correcta separación de capas de código.

Gravedad

Alta

Nomenclatura inconsistente

Situación

- Nomenclatura inconsistente de variables, clases, archivos, etc.
- Nombramiento inconsistente de archivos (uso de mayúsculas y minúsculas)

Riesgo

El no contar con un estandar de nomenclatura de variables dificulta la legibilidad del código y el mantenimiento del sistema.

Esto se traduce a mayores tiempos y costos ante cambios necesarios en el sistema.

Recomendación

Se recomienda utilizar un estándar de nomenclatura para las variables del código.

Gravedad

Media

Mezcla de lenguajes utilizados en un mismo archivo de presentación

Situación

Se esta mezclando varios lenguajes en un mismo archivo (html, php, js, css).

Riesgo

El combinar distintos lenguajes en un mismo archivo impide la segregación de tareas entre más de un programador además de fomentar la duplicidad de código, dificultar el mantenimiento y la legibilidad del código.

Esto se traduce a mayores tiempos y costos ante cambios necesarios en el sistema.

Recomendación

Las buenas práctica de desarrollo propone la separación de distintos lenguajes (Javascript, CSS, HTML) en archivos separados para facilitar su mantenimiento y segregar las labores en equipos de trabajo, facilitando así su mantenimiento.

Gravedad

Media

Problemas con Javascript

Uso de sintaxis inconsistente

Situación

Muchas líneas de código javascript no terminan con punto y coma ';' lo cual hace que el código no sea uniforme y puede causar problemas en ciertas situaciones.

Gran parte del código se encuentra probablemente indentado, lo que hace muy difícil su entendimiento y mantenimiento.

Riesgo

El no hacer un uso correcto de la sintaxis del lenguaje lleva a problemas de evaluación y la aparición de bugs difíciles de reparar.

Además de generar problemas al momento de comprimir el código para reducir el tamaño de las páginas.

Recomendación

Se recomienda definir estándares de programación de Javascript que aseguren una correcta codificación.

Gravedad

Alta

Uso de métodos no estándar

Situación

Se utilizan shortcuts que solo funcionan en Internet Explorer haciendo que el sistema sea incompatible con otros navegadores (document.element_name).

Riesgo

El utilizar llamadas a métodos no estándar dificulta las labores de migración a navegadores más modernos y afecta la seguridad de la empresa al forzar a los usuarios a utilizar versiones desactualizadas (2001) de las herramientas de acceso a Internet.

Recomendación

Se recomienda hacer uso de Frameworks de Javascript que abstraen las incompatibilidades de navegadores permitiendo así la migración de la empresa a navegadores más actualizados y seguros.

Gravedad

Alta

Inclusión y definición de funciones múltiples

Situación

Algunos scripts o funciones son incluidos mas de una vez produciendo warnings en el navegador (el calendario).

Algunas funciones javascript están repetidas en diferentes archivos.

Riesgo

El contar con duplicidad de carga de funciones indica una duplicidad de código lo cual genera problemas al momento de ejecutar funciones definidas previamente en Javascript.

Además de incrementar innecesariamente los tiempos de carga de las páginas entorpeciendo la rapidez del sistema.

Recomendación

Se recomienda hacer un análisis de pantallas para identificar que scripts están siendo cargados por duplicado y eliminar dichas lineas.

Gravedad

Media

Archivos mezclados en directorios

Situación

Los archivos javascript estan en la misma carpeta que los archivos php, haciendo difícil ubicarlos y mantenerlos.

Riesgo

El no tener una adecuada separación de los archivos de acuerdo al lenguaje dificulta el mantenimiento en equipos de trabajo además de no ser considerado una buena práctica de programación.

Recomendación

Se recomienda crear un directorio destinado a separar los archivos Javascript utilizados en las páginas

Gravedad

Baja

HTML

Documentos HTML declarados sin Doctype

Situación

No se está definiendo el doctype del documento.

Riesgo

El riesgo de no definir un doctype adecuado a los documentos HTML implica que el navegador funcionará en modo "quirks" lo cual entorpece los tiempos de desarrollo al tener que aplicar tiempo en el desarrollo para situaciones particulares de cada navegador además de imposibilitar el uso del sistema en otros navegadores.

Recomendación

Se recomienda definir un doctype y aplicarlo a lo largo del sistema.

Gravedad

Media

No se ha definido un encoding para los documentos

Situación

No se está definiendo el encoding del documento.

Riesgo

El no definir un encoding del documento afecta la manera en que se visualizará la información en distintos sistemas. Ocultando así algunos caracteres dificultando la lectura.

Recomendación

Se recomienda agregar la etiqueta <META/> a la cabecera de todos los documentos HTML.

Gravedad

Baja

Separación incorrecta de presentación de datos

Se están utilizando atributos no recomendables en algunos tags como bgcolor (esto debería hacerse por css).

Riesgo

El riesgo de no tener una correcta separación de presentación y documento complica las labores de rediseño y mantenimiento visual de la aplicación al tener que cambiar archivos por separado dentro de cada documento .html en lugar de en una hoja de estilos centralizada .css .

Recomendación

Se recomienda separar la definición visual de la página en archivos CSS para su fácil mantenimiento.

Gravedad

Baja

HTML no válido

Uso de HTML Inválido con etiquetas mal cerradas y estructuradas.

Riesgo

El uso de HTML no validado genera problemas en la programación Javascript dado a que la definición del DOM será asignada arbitrariamente por el navegador que puede diferir de la conceptualizada por el programador.
Esto se traduce a mayor tiempo de desarrollo y lenta reacción a cambios de los requerimientos.

Recomendación

Se recomienda realizar validaciones de HTML en las pantallas y asegurarse que todas las etiquetas se encuentren correctamente utilizadas

Gravedad

Baja

CSS

Referencias a archivos inexistentes

Existen referencias a archivos css inexistentes.

Riesgo

El hacer referencias a archivos inexistentes implica que el navegador tendrá que comunicarse innecesariamente con el servidor de aplicación web resultando en mayores tiempo de carga de página.

Recomendación

Se recomienda identificar las referencias a archivos inexistentes y eliminarlos de los documentos HTML que los invoquen.

Gravedad

Bajo

CSS inválido

Uso de CSS inválido

Riesgo
























El uso de sentencias de CSS inválido dificulta el mantenimiento de la línea visual debido a que habría que trabajar con una estrategia de error y corrección en lugar de poder seguir los estándares CSS.












Recomendación

Se recomienda realizar evaluaciones por herramientas de validación de CSS a los archivos de estilos de tal forma que se asegure la validez de dichos documentos.

Gravedad

Baja

Punto revisado	Criticidad
Revisión de base de datos	
Nombramiento inconsistente de tablas y columnas	Media 
Motor sin soporte de transacciones	Baja 
Uso de Charset con problemas de compatibilidad	Baja 
Tablase en desuso	Baja 
Uso de llaves primarias ineficientes	Alta 
Revisión de seguridad	
Mal uso de funciones de control de acceso	Alta 
Problemas de Inyección de SQL	Alta 
No se cuenta con protección adecuada para las funcionalidades	Alta 
Revisión de eficiencia de carga	
Lentitud en los reportes	Media 
Uso de SQL ineficiente	Media 
Uso de funciones específicas del motor de base de datos	Baja 
Uso excesivo de consultas en los reportes	Alta 
Revisión de usabilidad	
Inconsistencia en las notificaciones de error	Baja 
Los usuarios no están al tanto de la opción de cambiar contraseña	Media 
Pantallas confusas y sobrecargadas	Media 
Estructuración desordenada de las funcionalidades	Baja 
No existe información de ayuda del uso de las funcionalidades	Baja 
Controles mal ubicados	Media 
No existen mantenimientos	Alta 
Problemas de codificación	
No se ha comentado el código	Media 
Código redundante	Alta 
Mala indentación del código	Baja 
No hay separación de las capas de lógica	Alta 

Nomenclatura inconsistente	Media 
Mezcla de lenguajes utilizados en un mismo archivo de presentación	Media 
Uso de sintaxis inconsistente	Alta 
Uso de métodos no estándar	Alta 
Inclusión y definición de funciones múltiple	Media 
Archivos mezclados en directorios	Baja 
Documentos HTML declarados sin doctype	Media 
No se ha definido un encoding para los documentos	Baja 
Separación incorrecta de la presentación de datos	Baja 
HTML inválido	Baja 
Referencias a archivos inexistentes	Baja 
CSS inválido	Baja 