**Tecnológico de Costa Rica**

**Customer Relationship Manager**

**Reporte de auditoría de código**

**Version 1.3**

**Índice**

|  |  |  |
| --- | --- | --- |
| **Información del documento** |  | **3** |
| Descripción |  | **3** |
| Identificación del documento |  | **3** |
| Organización encargada |  | **3** |
| Historial de cambios |  | **3** |
| **Introducción** |  | **4** |
| Alcance |  | **4** |
| Referencias |  | **4** |
| Glosario |  | **4** |

|  |  |  |
| --- | --- | --- |
| **Desviaciones con respecto al estándar** |  | **4** |
| **Observaciones de la implementación**  **Recomendaciones y acciones a tomar** |  | **5**  **6** |

**1. Información del documento**

**1.1. Descripción**

El siguiente es un reporte que detalla el resultado de la revisión del código fuente del producto *Customer Relationship Manager*, referido de ahora en adelante como CRM.

Versión del software a revisar: 1.3

**1.2. Organización encargada**

Los siguientes recursos componen la organización encargada de llevar a cabo la inspección pertinente.

● Melissa Molina Corrales

● Jonnathan Rodríguez

● Edwin Cen Xu

**1.3. Historial de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción de los cambios** | **Autor** |
| 12/11/2017 | 1.3 | Primera versión | Jonnathan Rodríguez |
| 13/11/2017 | 1.3 | Se especifican algunas desviaciones con respecto al estándar. | Edwin Cen Xu |
| 15/11/2017 | 1.3 | Se especifican desviaciones con respecto al estándar y observaciones de implementación. | Melissa Molina Corrales  Jonnathan Rodríguez |

**2. Introducción**

**2.1. Alcance**

El propósito de este documento es proporcionar un informe resultante del proceso de reunión e inspección de código fuente llevado a cabo en el producto de software CRM. Durante este reporte, se llevará a cabo una revisión de la implementación de los conjuntos de funciones para el Portal de Clientes implementados:

* Reportes de errores
* Seguimiento a entrenamientos.
* Productos Relacionados

**2.2. Referencias**

[DEP] Documento de estándar de programación

**2.3. Glosario**

Las siguientes abreviaciones son usadas en el presente documento: CRM Customer Relationship Manager

DRY Don’t Repeat Yourself

**3. Desviaciones con respecto al estándar**

* En el inicio de cada archivo existen includes innecesarios.

● Igualmente se maneja de manera inconsistente el nombramiento de funciones y variables ya que a veces estos aparecen en español y otras en inglés.

* En todos los archivos existen comentarios donde no se aplica el espacio entre el delimitador de comentario y el texto del comentario.
* En el archivo ReportesErroresVendedor.aspx existe una función llamada Page\_Load la cual no se está utilizando y dentro de ella sólo existen comentarios.

● Hay archivos no utilizados dentro de la carpeta del proyecto, residuales de la plantilla utilizada en la carpeta *pages* y en la raíz del proyecto.

● Se presenta código duplicado al inicio de cada página, ya que reescribe el encabezado de cada una, lo que viola el principio DRY.

● Según el estándar [DEP], la manera de concatenar cadenas de caracteres es con la función nativa *Append()*, la cual no es usada.

● Faltan comentarios que indiquen cómo se está llevando a cabo la validación del formato de correo en el registro de usuarios y contactos.

● Se hacen operaciones con AND y OR sin paréntesis.

● Existen archivos que ya no se están utilizando porque sus funciones fueron trasladas a otro archivo por ejemplo el archivo: AutenticaciónCliente.aspx, les funciones de este archivo fueron trasladadas al archivo Autenticacion.aspx.

**4. Observaciones de la implementación**

● Falta validar la cantidad de filas recuperadas en la sección de ingreso al sistema. Esto para evitar problemas con inyecciones SQL.

● La función de ingreso al sistema refresca la interfaz si falla la validación de datos. Esto es innecesario y lleva a la pérdida de información.

● Muchas funciones generadas automáticamente por el IDE se mantienen en el código y no son utilizadas.

* Algunas funciones de verificación de datos son muy grandes y pueden simplificarse en muchas más pequeñas.

-Vease verificarDatosUsuario() en el archivo RegistroCliente.aspx

* El archivo RegistroCliente.aspx implementa las mismas funciones que usa el archivo Registro.aspx, provocando repetición de código.

● Mala implementación del registro de usuarios con la base de datos. El sistema crea dos conexiones distintas para realizar las transacciones, lo cual no es eficiente ni claro para el programador.

● Algunas interacciones con las consultas no son atómicos, lo que hace al sistema menos tolerante a fallos.

● Desacoplar código del cliente y del servidor. En algunas secciones se incluye código HTML en archivos de control.

● Se pueden encontrar nombres y títulos residuales del template utilizado

**5. Recomendaciones y acciones a tomar**

1. Revisar y renombrar todas las variables y archivos que están escritos en inglés a nombres en español.

2. Eliminar la carpeta *pages* y todos los archivos *.html* del proyecto.

3. Apegarse al principio DRY, esto implica crear archivos de encabezado y pie de página que contenga el mismo código con necesario para las distintas páginas así como las referencias a las bibliotecas importadas.

4. Usar los métodos nativos del lenguaje para realizar acciones sobre cadenas de caracteres.

5. Asegurarse de validar la cantidad de filas recuperadas en la consulta que autoriza la entrada al sistema.

6. Eliminar o darle uso a las funciones autogeneradas por el IDE.

7. Dividir las funciones en otras más pequeñas cuando sea aplicable.

8. Cerrar la conexión con la base de datos una vez terminadas las consultas en las funciones que interactúan con la base de datos y comprimir la funcionalidad de la duplicidad de los objetos de conexión a un único objeto reutilizable.

9. Hacer *commit* de los datos de las consultas con las bases de datos inmediatamente antes de cerrar la conexión.

10. Mantener el código HTML en archivos aparte que se puedan consultar para hacer la respectiva inyección del código al cliente.

11. Eliminar las funciones que no se estén utilizando en el código y no dejarlas con comentarios.

12. Evitar la repetición de código, si existen funciones que se pueden reutilizar.

13. Eliminar archivos que no se estén utilizando.

14. Poner solo los includes necesarios para cada clase.