**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** |
| 18/11/2017 | 1.3 | Primera versión | Edwin Cen Xu |
| 19/11/2017 | 1.3 | Se especifican algunas desviaciones con respecto al estándar. | Jonnathan Rodríguez |
| 20/11/2017 | 1.3 | Se especifican desviaciones con respecto al estándar y observaciones de implementación. | Melissa Molina |

**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**

● Algunos métodos tienen nombres que inician con mayúscula y otros con minúscula dentro de las clases.

* En el archivo RegistroCliente.aspx la función loginRef() no se utiliza.
* En el archivo RegistroCliente.aspx la función registrarUsuario() tiene comentarios de tipo /\* \*/ y comentarios que no tienen espacio entre el delimitador de comentario y el texto del comentario.

● Hay algunos nombres poco significativos, principalmente en la clase *GUIBuilder.*

● Algunos nombres de métodos son singulares y otros plurales, incluso dentro de la misma clase.

● Hay comentarios con código residual, principalmente en los métodos *Page\_Load* de cada página.

**4. Observaciones de la implementación**

● Hay código muy parecido entre métodos que podría agruparse en uno mismo. Esto es muy evidente en los métodos *autenticacion(), autenticacionCliente(), ingresar() e ingresarCliente().*

● Las clases de prueba, en la carpeta *Test*, deberían de ser separadas del proyecto principal y probadas en un proyecto aparte dedicado únicamente a ejecutar pruebas del sistema.

* En el archivo Verificador.cs la función verificarLargoContrasena() se debe dividir en verificar si la contraseña es corta y verificar si la contraseña es larga ya que al ingresar una contraseña corta o larga el sistema mostrará el mensaje “Largo de contraseña corto, digite 6 o más caracteres para su contraseña” para ambos casos y debe mostrar un mensaje diferente para el caso de que la contraseña sea muy larga.

**5. Recomendaciones y acciones a tomar**

● Alinear todos los nombres de métodos con el estándar, o sea, cambiarlos para que estos empiecen todos con letras minúsculas y sean singulares. Igualmente intentar hacer más significativos los métodos mencionados y escribir todos en un mismo idioma, ya sea español o inglés.

* Eliminar los comentarios con código residual para evitar confusiones al mantener el código.

● Tratar de evitar la repetición de código.

* Corregir el mensaje que aparece cuando un usuario se registra correctamente para que este sea más significativo.
* Aislar las pruebas y todo lo relacionado a estas en un proyecto distinto.

● Se sugiere cambiar el patrón de diseño. Debido a las ventajas que ofrece el *framework, ASP.Net,* se sugirió considerar la idea de reimaginar el sistema bajo las estructura MVC. Esto para ayudar a reducir la redundancia de código y aprovechar al máximo las funciones de la herramienta de desarrollo. También el cambio tendría la intención de reducir la cantidad de pruebas automatizadas necesarias para cubrir la mayoría del código.