

EMSYS

Informe de Revisión de código

Versión 12.2

Historia de revisiones

Fecha	Versión	Descripción	Autor
02/11/2016	12.1	Revisión BackEnd	Bruno Amaral
05/11/2016	12.2	Revisión FrontEnd	Bruno Amaral

Contenido

1. Producto revisado	3
1.1. Nombre y Versión del Producto revisado	3
1.2. Participantes de la revisión	3
1.3. Técnica utilizada	3
2. Objetivos de la revisión	3
3. Problemas detectados	3
3.1. Backend	3
3.2. Frontend	6
4. Evaluación	6
4.1. Acciones a tomar	6
4.2. Próxima Revisión del Producto	7

1. Producto revisado

1.1. Nombre y Versión del Producto revisado

Se revisó la rama dev del repositorio de **backend** siendo la siguiente la última actualización:

```
commit 073f14dbad52138c17714b97dee9532f36ad5701
Author: Andres Veiro <andress_08@hotmail.com>
Date: Wed Nov 2 01:56:01 2016 -0300
```

Se revisó la rama dev del repositorio de **frontend** siendo la siguiente la última actualización:

```
commit 3df1b0745164614186a941da8491f506223ac8ec
Merge: a4bb7da 2d5ec7b
Author: Marccio Silva <marccio.silva@fing.edu.uy>
Date: Sat Nov 5 00:10:23 2016 -0300
```

1.2. Participantes de la revisión

Brunos Amaral (Responsable de SQA)

1.3. Técnica utilizada

Se tiene un estándar propio de implementación establecido en el documento Estándar de Implementación que es una conjunción de requerimientos específicos del cliente y estrategias definidas por el equipo, para controlar que se cumpla con lo establecido en el documento se utilizan en ambos subsistemas del proyecto, backend y frontend, dos plugins stylecop y QAplug respectivamente, para complementar esto se harán estas revisiones de código donde se recorrerán los archivos de implementación buscando incongruencias no detectadas por las herramientas y además comunicar a los implementadores las falencias que sí se detectaron pero no han sido remediadas por estos.

2. Objetivos de la revisión

Mejorar la calidad del código es el principal objetivo. Además de por todos los beneficios que la calidad trae el cliente hizo énfasis en que se hará cargo de mantener el código entregado y se entiende que el hecho de que la calidad sea óptima colaborará con esto.

3. Problemas detectados

Dado que algunos de los errores encontrados en revisiones anteriores no fueron solucionados se procedió a identificar archivo y línea donde se detectan problemas para que la corrección sea realizada de forma más rápida por los implementadores.

3.1. Backend

Referencias de errores y sus sugerencias

A	Métodos y propiedades comentados si se van a implementar después poner ToDo sino borrar.
---	--

B	No se respeta lower camel case, si se trata de un atributo que viaja en Dto usar anotación JsonProperty(PropertyName....
C	Métodos sin firma o con firma incompleta.
D	Nombre de elemento con error de sintaxis.
E	Las variables privadas deben nombrarse iniciando con “_” .

Clase	Error	Línea/s
DtoEvento.cs	A	48,49
Configuration.cs	A	13, 41-44, 53-56, 71-74, 86-88, 178, 213-220, 238, 266-268
Evento.cs	A	26
Extension_Evento.cs	A	21, 24
DtoGetters.cs	C	14-343
TieneAcceso.cs	C	9, 46, 75, 99
TokenGenerator.cs	C	7
IMetodos.cs	B C	66, 79, 98, 120 97, 120, 188, 198, 208, 276, 285
Metodos.cs	B	329, 335, 366, 372, 403, 410, 446, 452, Unreachable code 486
Program.cs	B D	28 ,31, 34, 37, 40, 43 10
ProcesoAsignacionRecurso.cs	B C E	74, 80, 113 74 15, 17
ProcesoAsignacionRecursoDescripcion.cs	A B E	123-126 72, 78, 111, 15
ProcesoAudios.cs	B E	74, 80, 113 15, 17
ProcesoEventos.cs	A B D E	58 , 60, 84, 117 15 21
ProcesoExtensiones.cs	A	63

	B D E	86, 119 15 21
ProcesoGeoUbicacion.cs	A B E	120-142 78, 111, 117 15
ProcesoImagenes.cs	B E	80, 113 15, 17
ProcesoVideos.cs	B E	80, 113 15, 17
MyDateTimeConverter.cs	C	9, 14, 19
Startup.OAuth.cs	A	16-40
CustomAuthorization.cs	E	12
AdjuntosController.cs	A C	79-105, 250, 257, 272, 277, 282, 288, 310, 317, 329, 333, 338, 343, 349, 381,-386, 435-440, 489-495 66, 142, 191,
EventosController.cs	B	55
LoginController.cs	C	165
Startup.cs	E	17
FactoryNotifications.cs	A E	23-3, 42 15
INotifications.cs	A	5, 6
NotificationsFirebase.cs	A C E	38, 103, 60 20, 22

El estándar de implementación brindado por sonda establece que los errores se loguen de la forma

```
LogMaker.MakeErrorLog(ActionType.Y, ex, "En NOMBRE_FUNCION", EntityType.Z, ModuleType.W, (-1|id));
```

Sin embargo en el proyecto los errores se están logueando de la siguiente manera:
dbAL.AgregarLogError(token, "", "Emsys.ServiceLayer", "EventosController", 0, "ListarEventos", "Hubo un error al intentar listar eventos de un usuario, se adjunta excepcion: " + e.Message, MensajesParaFE.ErrorListarEventosCod);

Evaluar si realizar el cambio para que el formato sea el adecuado insumiría en un costo inmanejable en caso contrario realizar los cambios correspondientes.

3.2. Frontend

En la siguiente sección se mencionan algunos ejemplos de errores que se repiten en otros archivos para localizarlos todos ejecutar plugin de QA.

MultimediaUtils.java línea 31 se guarda variable local para después retornarla esto consume memoria de forma innecesaria mejor sería **hacer el return directamente**.

ExtensionsPresenter.java línea 10 modificadores redundantes, **eliminarlos**.

imports innecesarios en muchos archivos existen imports innecesarios recordar **eliminarlos**.

ExternalServiceResultActivity.java línea 63 se inicializa variable al final del método y no se usa puede ser indicio de un bug.

ImageGalleryPresenter.java línea 87 se declara variable se inicializa (String imageName = imageDataList.get(i).getName();) pero no se usa a lo largo del método sin embargo si se utiliza el valor pero accediendo a imageDataList.get(i).getName(), **eliminar declaración de variable o utilizarla en lugar de volver a acceder a dato**.

AttachGeoLocPresenter.java línea 25 se declara variable privada que no se usa nunca esto se repite en muchos archivos **ejecutar plugin y los que no se van a usar eliminar**.

EndpointService.java línea 38 se asigna un nuevo valor a un parámetro, **declarar variable local y asignar esta**, los parámetros no deberían alterarse.

AppRequestQueue.java línea 87 métodos vacíos sin comentar, **agregar ToDo**.

No se comentan las firmas de los métodos, **comentar la firma de los métodos explicando que hacen dichos métodos**.

4. Evaluación

No se encontraron problemas preocupantes salvo los requerimientos del cliente que no se están cumpliendo, todos los errores detectados se pueden solucionar sin que esto implique reestructurar el proyecto, se nota una gran mejoría comparado con las revisiones realizadas anteriormente.

4.1. Acciones a tomar

Corregir los errores encontrados lo antes posible y implementar siempre teniendo en cuenta los estándares de calidad.

Ejecutar plugins en ellos surgen advertencias que no vale la pena mencionar en este informe pero que aportan a la calidad del código.

4.2. Próxima Revisión del Producto

A fines de la semana 13 se tiene estipulada otra revisión.