**Trabajo Práctico 5.6**

**Seguimiento de desviaciones**

**Integrantes:**

**Emanuel Alberto Márquez.**

**Catedra:**

**Gestión de Calidad.**

**Carrera:**

**Licenciatura En Sistemas.**

**Docentes:**

**Mg. Albert Osiris Sofía.**

**Esp. Eder Dos Santos.**

**Año: 2020.**

**Fecha de presentación: 01 de Mayo de 2020.**

INDICE

[*1.* *Evaluar la adecuación de las recomendaciones PSR1 y PSR2. Analizar código del proyecto LDS.* 1](#_Toc39315743)

[*2.* *Identificar tres aspectos más relevantes cumplidos.* 1](#_Toc39315744)

[*3.* *Identificar tres aspectos más relevantes no cumplidos.* 3](#_Toc39315745)

[*4.* *Proponer una adecuación.* 6](#_Toc39315746)

1. *Realizar una evaluación de distintas herramientas open source de seguimiento de incidentes (Issue Tracking) y errores (Bug Tracking).*

**Bugzilla**

Es un sistema de seguimiento de errores. Como tal, permite a los equipo de desarrollo de software realizar seguimiento a los errores, problemas, mejoras y otras solicitudes de cambios en los productos de una forma efectiva. A pesar de ser gratuito, tiene muchas características que son capaces de competir con sus contrapartes pagas. Para poder ser utilizado, debe ser instalado en un servidor.

**GLPI**

Es un sistema de seguimiento de incidencias que se encuentra desarrollado en PHP bajo una distribución GPL. Es una aplicación web que permite a las organizaciones construir un inventario de los recursos y de gestionar tareas administrativas. Sus funcionalidades ayudan a administrar activos y de proporcionar historiales de intervenciones.

1. *Identificar tres aspectos más relevantes cumplidos.*

**PSR1 – Constantes de clases, propiedades y métodos**

Si bien la guía evita hacer alguna recomendación sobre el nombre de propiedades, indica que debe mantenerse una consistencia a lo largo del proyecto. En el caso de Tempus, se establece el formato StudyCaps para los nombres de clase y camelCase para atributos y métodos.

1. *Identificar tres aspectos más relevantes no cumplidos.*

**PSR1 - Espacios de nombres y nombres de clases**

Según la recomendación, los espacios de nombres y los nombres de las clases deben seguir un PSR de “Autoloading”. Esto permite que al utilizar funciones de autoload, sea más eficiente la solicitud de REQUIRE o INCLUDE. Con tal objetivo, la clase debe indicar el espacio de nombre en al menos un nivel de la estructura.

Las constantes deben declarase en mayúscula con separadores de subrayado. Pero, además se recomienda declarar las constantes como parte de una clase y no como un simple archivo.

En la siguiente imagen observamos nuestro archivo de constantes:

1. *Proponer una adecuación.*

El código adecuado se adjunta como parte de la entrega. A continuación observamos algunas de las modificaciones incorporadas:

.