

Tienda Virtual

Documentación de pruebas estáticas

Versión 1.2

ID: PE-001

Versión: 1.2

Día: 28/10/2018

Autores: Luis Pablo Monge Angulo (LP)

Josué Vargas Hernández (J)

Índice

1. Información del documento	3
1.1.Descripción	3
1.2.Identificación del documento	3
1.3.Organización encargada	3
1.4.Historial de cambios	3
2. Introducción	4
2.1.Alcance	4
3. Desviaciones con respecto al estándar y resultados de pruebas	4
4. Observaciones de la implementación	5
5. Recomendaciones y acciones a tomar	6

1. Información del documento

1.1. Descripción

El siguiente documento es un reporte con las pruebas estáticas realizadas al código fuente del producto Tienda Virtual.

1.2. Identificación del documento

PE-001

1.3. Organización encargada

Los siguientes recursos componen la organización y roles encargados de llevar a cabo la prueba

- Luis Pablo Monge: Moderador, revisor y escriba para el módulo administrador. Como productor para el módulo de cliente.
- Josué Vargas como Moderador y revisor para el módulo cliente. Productor para el módulo administrador.

1.4. Historial de cambios

Fecha	Versión	Descripción de los cambios	Autor
28/10/2018	1.2	Primera versión	Luis Pablo Monge

2. Introducción

2.1. Alcance

El propósito de este documento es proporcionar un informe resultante del proceso de reunión y pruebas de las pruebas estáticas del código fuente del producto Tienda Virtual. Para el reporte, se llevará a cabo una revisión estática de la implementación de los requerimientos implementados hasta el momento de la prueba: módulo cliente y módulo administrador.

3. Desviaciones con respecto al estándar y resultados de pruebas

Para el estándar, se indicó que las funciones y variables estarán en español, evitando el uso de la letra ñ. Además que los nombres de las variables, funciones y clases deben estar en minúsculas, excepto la primera letra de cada palabra a partir de la segunda. Una vez explicado se procede a enlistarlos:

- Algunas declaraciones no siguen el estándar. Como la función *search* de *ProductoController*. Severidad: media
- El nombre de los controladores tiene dos palabras en idiomas distintos, a excepción de *AdminController*, que se llegó a la conclusión que no sigue ningún estándar por ser *Admin* una forma abreviada de Administrador u Administrator, sin dejar claro en qué idioma está. Severidad: media alta
- Sigue habiendo secciones de código que no llegó a ser cambiado por un refactor que se tuvo que hacer para dejar encargado al modelo la conexión con la base de datos. Severidad: media
- Hay funciones en el código que no se sabe si se usan. Severidad: alta
- En *ProductoController* hay líneas repetidas por función, se dijo que se podían agrupar en otra función para evitar repetir código, el productor alegó que debe ser así por claridad de código, si lo implementa por medio de una función, lo tendría que retornar como lista y pasarlo por partes a la vista y le quitaría claridad al código. Severidad: baja
- Hay funciones sin documentar, ya sea en *php* o *javascript*. Severidad: Alta

- Hay validaciones en *javascript* que también se hacen en *JQuery*. El productor alegó que *jQuery* dejó de responder y reimplementó las validaciones en *javascript* y no borró por medio a que afectase otras secciones. Severidad: media-baja
- El método *eliminarProducto* debería ser cambiado de nombre por el hecho que no elimina el producto de la base de datos, cambia su estado de habilitado a deshabilitado, provocando confusión. Severidad: media
- La clase *ProductoController* está sobre cargada. Los dos productores alegaron que todo lo que se tiene afecta a producto y no saben donde más poner esos métodos. Severidad: baja
- Falta validar en ciertas secciones del código, la cantidad de filas recuperadas de la base de datos para evitar inyecciones SQL. Severidad: alta
- No hay una validación de datos por parte del backend. Severidad: alta

4. Observaciones de la implementación

- La función de login al sistema refresca la interfaz si falla la validación de datos. Esto es innecesario y lleva a la pérdida de información.
- Registrar usuario crea el *query* necesario para la lógica de la base ahí y de forma muy compleja
- Muchos comentarios en el html.
- Hay páginas que todo el html está y otras en las que usa las plantillas *blade*. No hay un acuerdo de qué páginas usar plantillas y cuales no
- 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 de la plantilla utilizada
- En el css quedan muchas líneas que no son usadas por el hecho que la plantilla implementaba más de lo necesario y a la hora de quitar lo no relevante, no se validó esa parte.

5. Recomendaciones y acciones a tomar

1. Cambiar los de las funciones y clases mencionadas anteriormente
2. Revisar y renombrar todas las variables y archivos que están en escritos inglés a nombres en español.
3. Decidir si *AdminController* está en español o inglés y tomar la acción correctiva para cada caso.
4. Decidir si usar las plantillas *blade* que el *framework* ofrece para las vistas en todas las páginas.
5. Asegurarse de validar la cantidad de filas recuperadas en las secciones en las que esto lo amerite.
6. Validar los datos en la sección del backend para una mayor seguridad
7. Dividir las funciones en otras más pequeñas cuando sea aplicable.