ASH Web
Informe Final de Verificación del Sistema
Versión 1.3

Historia de revisiones

Fecha	Versión	Descripción	Autor
09/11/2016	1.0	Primera versión	Rodrigo Viera
11/11/2016	1.1	Actualización	Rodrigo Viera
13/10/2016	1.2	Revision SQA	Alvaro Callero

Contenido

1. Objetivos del documento

El objetivo de este documento es informar sobre las verificaciones realizadas, se pretende brindar una visión del desempeño de la verificación del Sprint 1, 2, 3 y 4. En las secciones siguientes se describen los resultados y el desarrollo de las pruebas realizadas durante estos primeros cuatro sprints.

2. Verificación de la implementación

2.1. Revisiones de código

En el equipo contamos con dos expertos en las tecnologías utilizadas, quienes fueron designados como responsables de desarrollo de backend y frontend. En estos Sprints que han pasado ellos en conjunto de algún integrante del equipo de verificación se han encargado de realizar las revisiones de código manualmente cada vez que un implementador realizaba un Pull Request en GitHub, es decir, cuando un implementador solicitaba que se haga un merge de la branch en la cual había estado desarrollando a la branch de staging. La branch de staging es la que utiliza el equipo de verificación para realizar las pruebas del sistema.

Siempre que se detectó un error al momento de realizar la revisión de código, se le notificó al implementador que realizó el Pull Request (PR) marcando los errores que fueron encontrados para que éste los corrija y vuelva a hacer el PR. De esta forma, se logra tener una branch mas estable para probar y un código más prolijo.

A su vez, se utiliza una herramienta llamada TravisCI que permite ejecutar ciertos scripts cada vez que los implementadores realizan un push en la branch principal para integrar lo que han desarrollado. Estos scripts son Reek, Rubocop y Rails Best Practices, los cuales permiten obtener un código más estable y unificado. Es decir, se encargan de analizar el código con ciertas reglas preestablecidas como largo de las sentencias y funciones, detectar posibles inconsistencias como métodos duplicados, estructuras ineficientes, condiciones repetidas, etc.

2.2. Pruebas unitarias y cobertura de código

Hasta el momento se han realizado pruebas unitarias en backend y estas fueron realizadas por parte del equipo de verificación (Isabel Ivagnes y Rodrigo Viera). Con la realización de dichas pruebas se obtuvo un cubrimiento final del 96,57% con un total de 69 pruebas exitosas y 0 fallas.

3. Verificación de Sistema

3.1. Pruebas funcionales

Se realizaron pruebas de caja negra de todas las funcionalidades implementadas por parte del equipo de verificación. Dicho equipo, en conjunto con los implementadores, decidió utilizar el gestor Trello para reportar bugs y para realizar sugerencias de mejoras. De esta forma, luego de finalizar la ejecución de las pruebas, todos los bugs detectados se clasifican en sí corresponden al equipo de

backend o al equipo de frontend, lo cual agiliza el procesamiento de los bugs y que estos sean resueltos de una manera más ágil. Además, los bugs también son clasificados en diferentes categorías de severidad (Crítico, Marginal y Menor). En el gestor de tareas Trello se creó un tablero con el nombre "Bugs" y ahí es donde el responsable de verificación reporta los errores encontrados luego de clasificarlos. Los responsables de desarrollo asignan los bugs a los implementadores y cuando estos los resuelven, los "mueven" hacia otro tablero para que el equipo de verificación se encargue de verificar que se haya solucionado el error. Una vez aprobado, éste es archivado.

Dentro de las pruebas de caja negra se realizaron pruebas de regresión y algunas pruebas de caja negra sin documentar los casos probados. Para ambos tipos de pruebas se realizó tanto un testing de funcionalidades como una combinación de ellas. En las pruebas se trató de verificar el correcto funcionamiento de la aplicación así como también el aspecto visual y la amigabilidad.

4. Verificación de Documentos

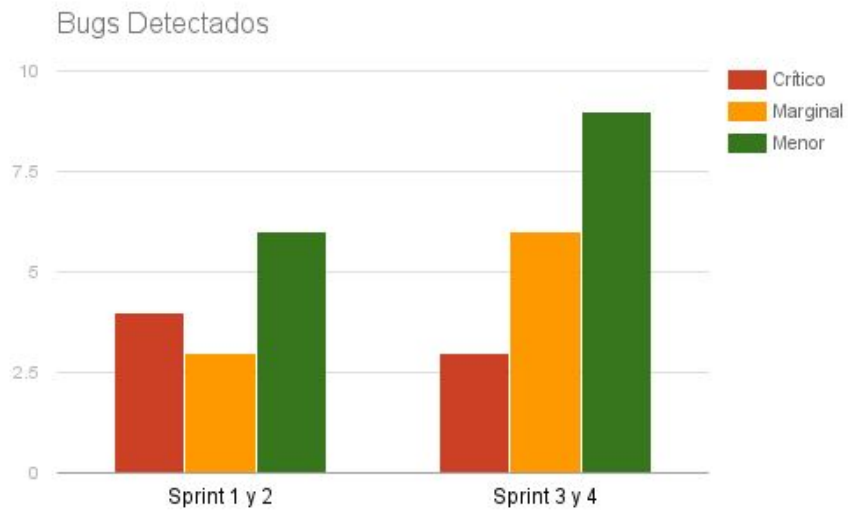
La verificación de los documentos a ser entregados es realizada a medida que los documentos estén disponibles y cada vez que se realice una modificación sobre el mismo. Para las entregas de cada Sprint se mantiene un plazo de 2 horas previo a la entrega para disponer de los documentos.

La verificación de los documentos es realizada por parte del Responsable de SQA Alvaro Callero.

5. Estadísticas

5.1 Bugs Detectados

A continuación se presenta el histórico de bugs detectados a lo largo de los 4 primeros Sprints.



Durante la verificación del Sprint 1 y 2 se registraron 4 errores críticos, 3 errores marginales y 6 menores. Para la verificación del Sprint 3 y 4 se registraron 3 críticos, 6 marginales y 9 menores.

Algo importante a resaltar es que durante los dos primeros sprints la mayoría de los errores se encontraron en backend. Para la segunda verificación, los errores de backend fueron solucionados en su totalidad, mientras que los errores de frontend no fueron corregidos en su totalidad debido a que presentaban problemas de tiempos para finalizar las cards que tenían asignadas y además eran pocos errores.

Para la segunda verificación, la mayoría de los errores fueron detectados en frontend. Por lo cual se decidió en conjunto con el cliente utilizar un Sprint para la corrección de los errores y ajustar algunas funcionalidades y así dejar el sistema lo más estable posible.

5.2 Bugs Corregidos

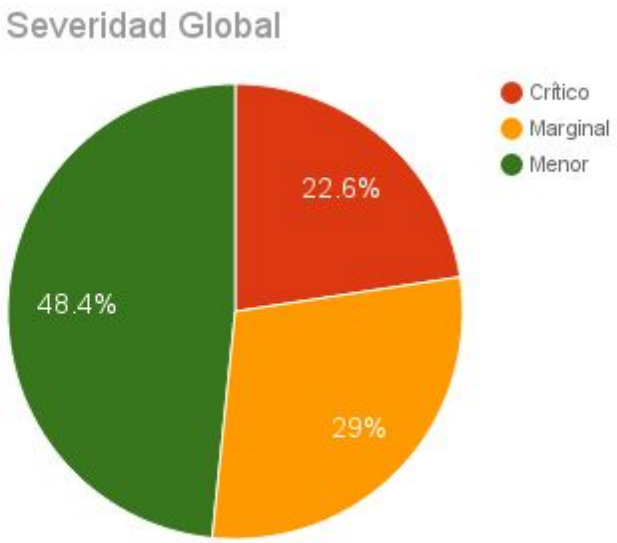
A continuación se presenta el histórico de bugs corregidos a lo largo de los 4 primeros Sprints.

Como es posible observar, de los errores encontrados durante el Sprint 1 y 2 se lograron corregir 12 errores de un total de 13. El único error que no pudo corregirse a la semana siguiente de finalizada la verificación fue un error marginal que se lo postergó una semana más. Para la verificación del Sprint 3 y 4 se lograron corregir todos los errores entre el Sprint 5 y 6 ya que el Sprint 6 básicamente se utilizó para solucionar errores y dejar el sistema lo más estable posible.

También se puede ver claramente que durante la segunda verificación se encontraron más errores que en la primera verificación y esto se debe a que las funcionalidades implementadas en el Sprint 3 y 4 fueron más complejas que las implementadas en el Sprint 1 y 2, que involucraban alta de usuario, cierre de sesión, iniciar sesión, envio de email y reseteo de contraseña.

Por último cabe resaltar que la cantidad de errores críticos no es alarmante ya que solamente se han encontrado 7 errores críticos para todas las funcionalidades complejas que se han implementado.

5.3 Severidad



En el gráfico anterior es posible observar la clasificación de los errores detectados a lo largo de estos Sprints que han transcurrido. Se puede apreciar que en el sistema se han detectado mayoritariamente bugs clasificados como menores.

Es de esperar que para las futuras verificaciones la cantidad de errores críticos disminuya debido a que el sistema presentará una estructura más estable.