

Plan de verificacion y validacion

NO Lines

Ver: 1.4

Federico Scuoteguazza

Martin Manasliski

Juan Tomas Tejeria

Propósito del documento

El propósito de este documento es detallar todas las actividades relacionadas con el plan de verificación y validación. El objetivo es lograr que los requerimientos sean realmente los pedidos por el cliente y que el software tenga una cantidad mínima de errores.

Se describirán las principales fases del proceso de validación y verificación. así como también se detalla las herramientas, recursos y procedimientos para realizar pruebas en el software.

Define, Quien va a ser responsable de llevar a cabo los tests, Que documentos van a ser necesarios, y Como se van a realizar.

Alcance

El plan va a estar dirigido fuertemente a aquellas actividades donde el equipo de trabajo tenga menos experiencia. intentando así evitar demoras a la hora de las entregas.

las grandes áreas a apoyarse serán:

- *Gestion*
- *Relevamiento de requerimientos*
- *Diseño*
- *Testing*

Organización

La organización del equipo de trabajo es:

Ver documento Roles 0.1

Gestion

La gestión es una de las actividades de mayor repercusión en el proyecto. Pudiendo generar grandes demoras en las diferentes entregas.

Al ser un equipo inexperto en esta área, es muy probable que ocurran errores, ya que depende mucho de la experiencia.

Para evitar los errores se realizará consultas a consultores más experimentados, reduciendo la probabilidad de que se cometan fallos.

También se requiere ir construyendo y actualizando documentos detallados más adelante, logrando una mejor gestión utilizando estos como ayuda.

Todos los documentos que entran en la gestión de configuración van a ser validados. una vez terminado el redactor le entregará el documento a otro integrante del equipo. este siguiendo un estándar designado para cada documento hará los cambios necesarios para que el documento mejore en calidad.

Relevamiento de requerimientos

Si bien también requiere de experiencia, es un proceso en el cual se puede verificar y validar los resultados antes de que generen grandes problemas. Por esta razón es que se van a tomar muchas medidas para que los problemas no crezcan.

Seguiremos el estándar IEEE-830 para la realización del documento de requerimientos, logrando crear un documento detallado sobre todos los requerimientos.

Todos los requerimientos serán verificados y validados por una persona distinta a quien los escribió. Una segunda opinión beneficiará en gran parte a que estén bien escritos.

Se verificará que los requerimientos sean necesarios, no ambiguos, completos, concisos, verificables y consistentes. Ya que son las características de todo buen requerimiento.

En caso de no llegar a un consenso en una discusión sobre un requerimiento se buscará que una persona externa resuelva la disputa.

1.Requerimientos para verificar

Se marcarán todas las funcionalidades y requerimientos que serán verificados.

Ordenados de mayor prioridad a menor según entendió el equipo a la hora de relevar requerimientos.

- 1.1
- 1.2
- 1.3
- 1.4
- 1.5
- 1.6
- 1.7
- 1.8
- 1.9
- 1.10
- 1.11
- 1.12
- 1.13

Requerimientos no funcionales que serán verificados

- 1
- 2
- 3
- 4

Objetivos de la Verificación

Los objetivos de la verificación son reducir la cantidad de errores en el software una vez hecho un release.

No se van a aceptar errores críticos en el programa que llega al público y la cantidad de errores intermedios y básicos serán reducidos al mínimo.

Estrategia de Verificación

Para realizar las pruebas, excepto las pruebas unitarias, se actualizaba a la última versión disponible en el repositorio, y luego se disponía a realizarse las pruebas correspondientes. una vez terminadas las pruebas, estas eran reportadas para arreglar en el futuro.

Tipos de Pruebas

En la siguiente sección se enlista los diferentes tipos de prueba a realizar.

2.1.1. Prueba de integridad de los datos y la base de datos

2.1.1.1. Objetivo de la prueba

Asegurar que los métodos no generen cambios indeseados en la base de datos.

2.1.1.2. Técnica

Ingresa diferentes datos “pilotos” y luego verificar de forma manual (con la herramienta que permite ver la base de datos) que hayan sido ingresados de manera correcta a la base de datos.

2.1.1.3. Criterio de aceptación

No generaron datos distintos a los esperados.

2.1.2. Prueba de Funcionalidad

Las pruebas de funcionalidad se dedican a verificar que cada requerimiento corresponden a casos de usos o funciones desarrolladas. y también comprobando que la implementación fue hecha correctamente

2.1.2.1. Objetivo de la prueba

Asegurar un funcionamiento correcto de cada función y método.

2.1.2.2. Técnica

Se ejecutará cada funcionalidad verificando que los resultados sean los

esperados.

2.1.2.3. Criterio de aceptación

Las Pruebas fueron realizadas con éxito.

2.1.3. Prueba de Performance

Se medirán y analizarán tiempos de respuesta, tiempo de espera en transacciones y cualquier requerimiento relacionado con algún tipo de interacción con el usuario.

2.1.3.1. Objetivo de la prueba

Notar si la aplicación tiene demoras no deseadas por el usuario.

2.1.3.2. Técnica

Una vez instalada la aplicación en un dispositivo, se harán recorridos por la aplicación notando los tiempos de respuesta.

2.1.3.3. Criterio de aceptación

El criterio de aceptación caerá sobre el tester designado a esto.

2.1.4. Pruebas de humo

Generar recorridas enteras del programa revelando errores que pasaron desapercibidos.

2.1.4.1 Objetivos de la prueba

Encontrar errores que quedaron cubiertos en pruebas anteriores.

2.1.4.2 Técnica

Antes de que el programa se encuentre listo para producción, se realizará estas pruebas para tener una referencia de la calidad del producto

2.1.4.3 Criterio de aceptacion

Será aceptado cuando se encuentren menos de 5 errores no críticos.

2.1.5. Pruebas Unitarias

Cada desarrollador está a cargo de sus pruebas unitarias ayudando a reducir el trabajo a la hora de hacer el testing, encontrando posibles errores en las primeras etapas.

2.1.4.1 Técnica

Se utilizará el módulo JUnit en java. En nivel de interfaz no se harán este tipo de pruebas ya que no se lo encuentra productivo

2.1.4.2 Criterio de aceptación

El módulo tendrá que pasar todas las pruebas.

Pruebas primera iteración

- Componentes

Los componentes verificados en nuestro proyecto serán la capa de persistencia la capa de negocios y la capa de interfaz.

En la capa de persistencia se verificará:

- Poder ingresar restaurants con sus respectivos datos a la base de datos.

Requerimiento asociado : 1.4

Persona encargada : Juan Tomas Tejeria

- Poder ingresar menús, asociados a un restaurante determinado.

Requerimiento asociado : a Agregar en próxima versión ESRE

Persona encargada : Juan Tomas Tejeria

Fecha de comienzo : Sábado 24 de Mayo

Fecha de finalización : Lunes 26 de Mayo

En la capa de negocio se verificará:

- La lista de restaurantes a mostrar es la adecuada.

Requerimiento asociado : 1.4

Persona encargada : Martin Manasliski

- La lista de menús de determinado restaurante es la correcta.

Requerimiento asociado : a Agregar en próxima versión ESRE

Persona encargada : Martin Manasliski

Fecha de comienzo : Sábado 24 de Mayo

Fecha de finalización : Lunes 26 de Mayo

En la capa de Interfaz se verificará:

- La lista de restaurantes es la adecuada y estos son botones seleccionables.

Requerimiento asociado : 1.4

Persona encargada : Federico Scuoteguazza

- La lista de restaurante es Scrollable

Requerimiento asociado : 1.4

Persona encargada : Federico Scuoteguazza

- La lista de menús de determinado restaurante se muestra con el formato requerido y cada línea es seleccionable.

Requerimiento asociado : 1.4

Persona encargada : Federico Scuoteguazza

- La lista de menús es Scrollable

Requerimiento asociado : 1.4

Persona encargada : Federico Scuoteguazza

Fecha de comienzo : Sábado 24 de Mayo

Fecha de finalización : Lunes 26 de Mayo