

PRUEBAS DE SOFTWARE

Sesión I del 26-05-2021

Instructor: Gerardo Mauricio Polania G

Ficha: 204989I TV ADSI TV CEET

QUE SON LAS PRUEBAS DE SOFTWARE

Las pruebas de software se han convertido en un factor determinante para lograr el éxito de sistemas, de ahí la importancia de establecer una metodología que nos guíe el camino para llegar a nuestro objetivo, un software con la mayor calidad posible. Aunando en este tema ¿qué es calidad? y ¿cómo podemos estar seguros que el sistema cubre todos y cada uno de los aspectos que conforman la calidad?, estas preguntas y otros temas relacionados son los que veremos en el contenido de este documento.

Las ventajas de establecer un proceso de calidad son:

- ▶ Detectar la mayor cantidad de errores posibles antes de salir a producción
- ▶ Ayudar a los administradores a la toma de decisiones
- ▶ Buscar los escenarios seguros para el uso del producto
- ▶ Evaluar la calidad
- ▶ Verificar la corrección del producto
- ▶ Asegurar la calidad

El presente es un trabajo en el cual se tratará el tema de pruebas de software y se definirán algunos aspectos importantes para así conocer el aseguramiento de la calidad y el proceso de pruebas que conlleva.

ATRIBUTOS DE LA CALIDAD DEL SOFTWARE

Como podemos observar lograr la calidad no es algo sencillo, el ISO 9126 dice que la calidad del software se divide en atributos funcionales y atributos no funcionales, así que para alcanzar la calidad debemos validar lo siguiente:

Atributos funcionales:

- Adecuación
- Exactitud
- Interoperabilidad
- Seguridad
- Cumplimiento de la funcionalidad

Atributos no funcionales de la calidad

- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad
- Fiabilidad

ACTIVIDADES DEL CICLO DE VIDA DEL DESARROLLO DE SW EN TEMAS DE CALIDAD

El conjunto de actividades de pruebas dentro del proceso de desarrollo de software, son conocidas como proceso básico de pruebas, el cual incluye:

Planeación. Donde se hace un esquema de ¿qué se va a probar?, ¿cómo se va a probar?, ¿quién lo va a probar? y ¿cuándo? A la salida de esta fase podemos mencionar que se obtiene un plan de pruebas maestro que incluye estrategia y enfoque de pruebas.

Análisis y diseño de pruebas. En esta fase se analizan los requerimientos y se diseñan los casos de prueba, incluyendo en esta actividad casos de prueba positivos y casos de prueba negativos, podemos encontrar más detalle de lo que debe contener un caso de prueba dentro del estándar IEEE 829. A la salida de esta fase se tendrán los casos de prueba.

Ejecución de pruebas. Esta fase es la más importante ya que, es donde los casos de prueba son ejecutados en un ambiente de prueba o calidad para validar que los requerimientos especificados se hayan implementado de la manera correcta, es en este punto donde se aplica la estrategia de pruebas. A la salida de esta fase tendremos el avance de ejecución y los reportes a directivos (generar información para las partes interesadas).

Evaluación de resultados. En esta fase es donde determinamos si se han alcanzado los objetivos de las pruebas, es decir, si la implementación de los requerimientos fue la óptima, lo ideal es que el usuario realice también otras pruebas para dar el visto bueno a la aplicación y ésta pueda ser puesta en producción.

Cierre de pruebas. Es la última fase del proceso de pruebas, aquí archivamos toda la documentación generada y se realiza una carta de aceptación de cierre que debe ser firmada por los directivos involucrados, también es bueno incluir los riesgos activos de la aplicación, si es que aplica.

ETAPAS TÉCNICAS Y TIPOS DE PRUEBAS

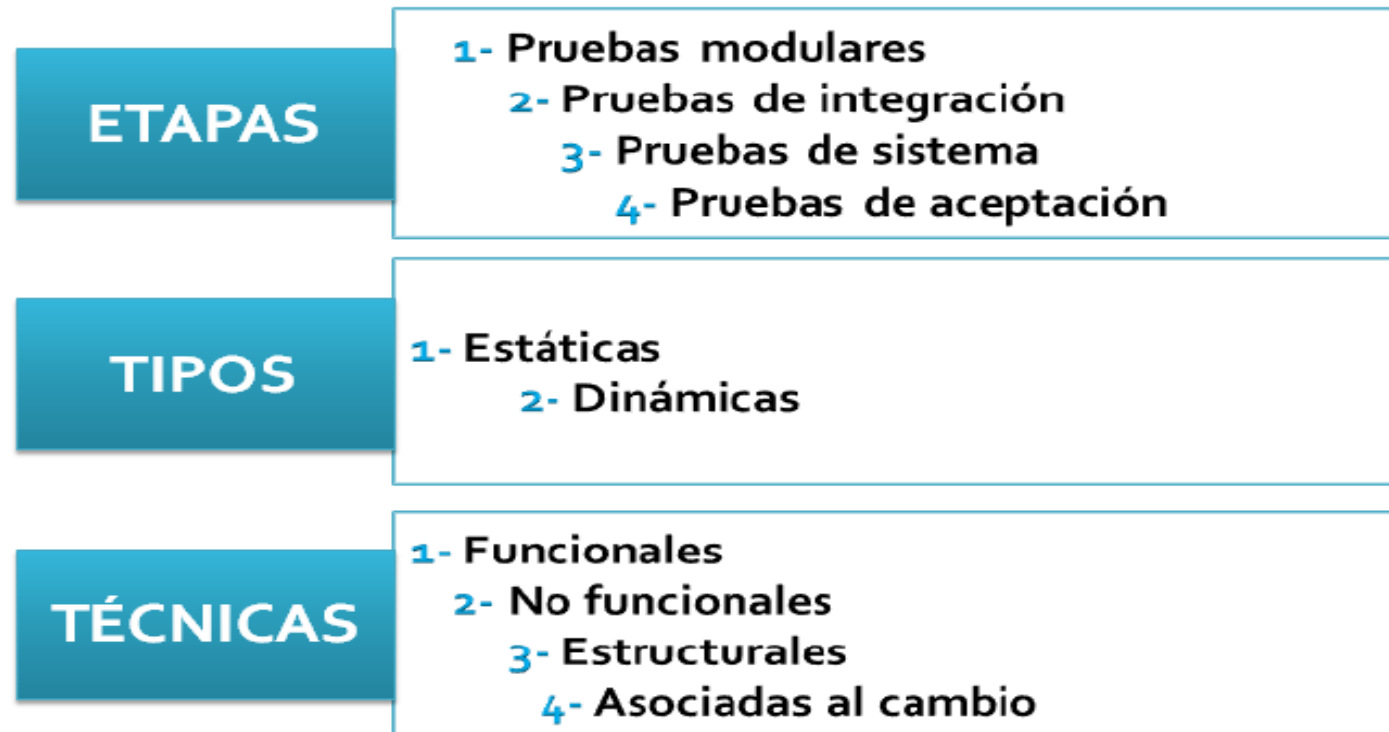


Figura 3.1 Etapas, tipos y técnicas de pruebas

PROPÓSITOS DE UNA PRUEBA

El propósito de las pruebas es:

- Detectar la mayor cantidad de errores posibles
- Ayudar a los administradores a la toma de decisiones
- Buscar los escenarios seguros para el uso del producto
- Evaluar la calidad
- Verificar la corrección del producto
- Asegurar la calidad
- Competitividad comercial
- Alcanzar mayor calidad en el software
- Cambios en la tecnología
- Reducción de costos y riesgos
- Incremento de productividad

EN RESUMEN 3 PUNTOS IMPORTANTES

- ✓ **Mejora de la calidad de un producto software.** El proceso de pruebas ayuda a suministrar o aportar al software los atributos deseados, por ejemplo: retirar defectos que conducen a fallos.
- ✓ **Reducción del riesgo de detectar errores.** Las actividades de pruebas de software adecuadas reducirán el riesgo de encontrar errores durante la fase de operación del software.
- ✓ **Satisfacer compromisos.** La ejecución de pruebas puede ser un requisito obligatorio por parte del cliente, debido a normas legales, así como al cumplimiento de estándares propios de una industria.

PUNTOS CLAVE PARA IDENTIFICAR UNA PRUEBA

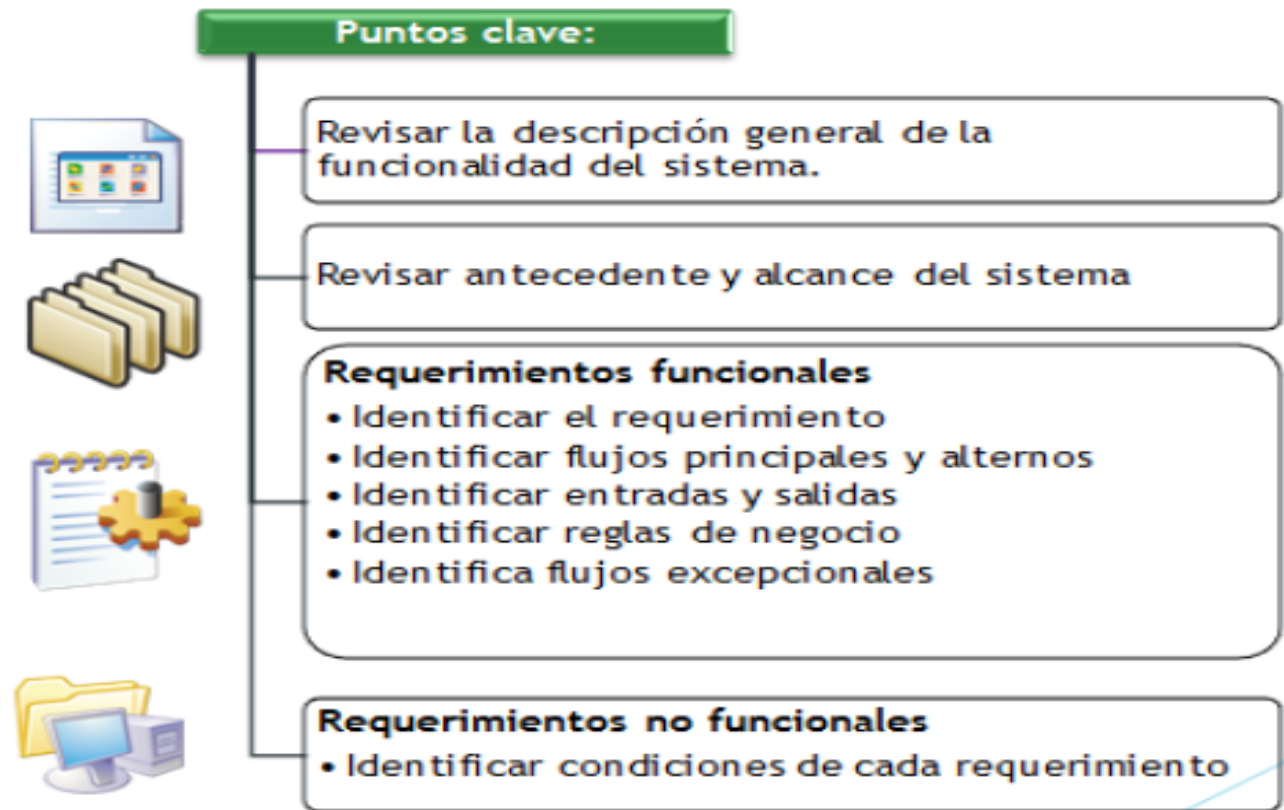


Figura 3.3 Identificación de requerimientos

CASO DE UNA PRUEBA

Definición: documentación que especifica las acciones a seguir para llegar a un objetivo específico (resultado esperado) que contiene o debería contener datos de entrada, resultados esperados y un conjunto de condiciones de ejecución de un elemento de prueba.

Lo que debe contener un caso de prueba según el estándar IEEE 829, es:

- Precondiciones
- Valores de entrada
- Resultados esperados
- Postcondiciones
- Identificador único
- Dependencia de otros casos de prueba
- Requisitos
- Forma en la cual se debe ejecutar el caso de prueba
- Prioridad

TIPOS DE CASOS

Tipos de casos

- **Casos de prueba positivos (Test To Pass):** Son aquéllos que se diseñan para probar el requerimiento tal cual se diseñó y construyó. Estos casos de prueba aseguran que las funciones descritas en los casos de uso o requerimientos funcionales realmente estén implementadas en el sistema o módulo que se esté probando y que funcionen como se definieron. Muestran funcionalidad.
- **Casos de prueba negativos (Test To Fail):** Son diseñados con la intención de romper el requerimiento, módulo o sistema a probar, es decir, comprueban situaciones en las que hay tratamientos con errores, por ejemplo: datos de entrada inválidos.
- **Happy path:** Escenario default de un requerimiento. No se consideran caminos alternos, excepcionales o condiciones de error.

MODELO DE VALIDACIÓN EN V

La validación se refiere a la corrección de cada nivel de desarrollo, de esta manera se comprueba lo adecuado de los resultados de un nivel de desarrollo.

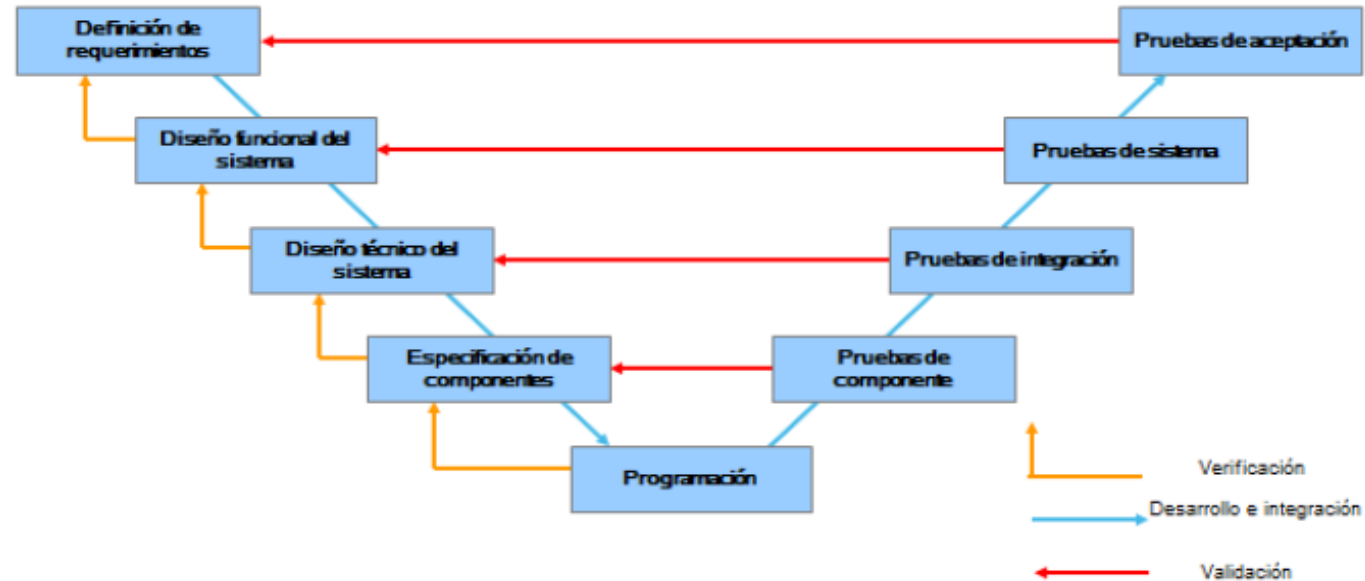


Figura 4.1 Modelo V

TÉCNICAS



COMO SE AGRUPAN

Tipos de pruebas

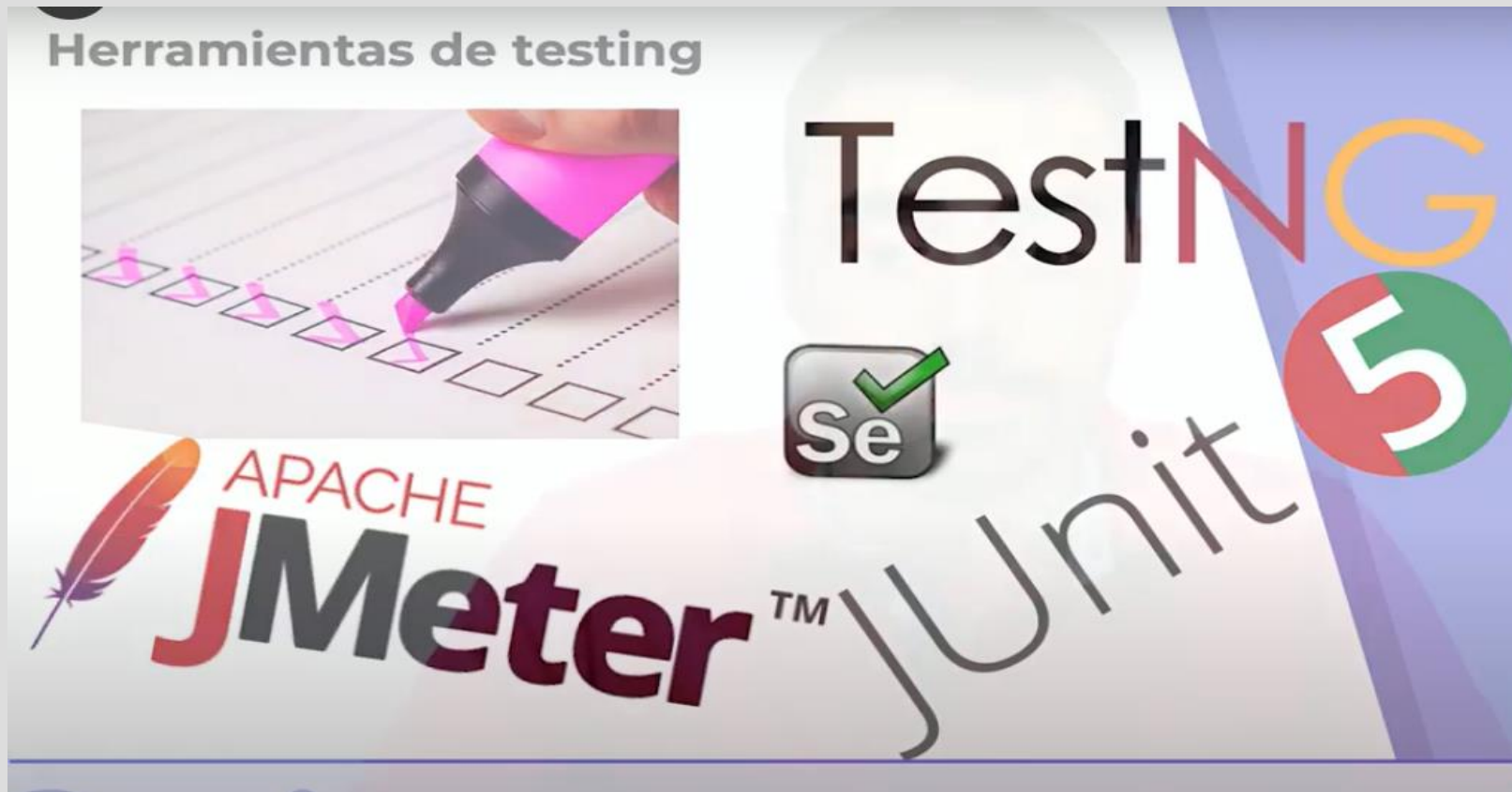


PRUEBAS DE CAJA BLANCA

PRUEBAS DE CAJA NEGRA



ALGUNAS HERRAMIENTAS DE TESTING



LISTA DE SOFTWARE PARA PRUEBAS

- **Listado de herramientas**
- Selenium (Web Application Testing)
- Appium (Mobile Testing)
- JMeter (Load Testing)
- Jenkins (Continuous Testing)
- TestLink (**T**est Management)
- Mantis (Bug-Tracking & Project Management)
- Postman (API Testing)
- Firebug / Firepath (Online Debugging)

Las hemos dividido en las siguientes categorías según sean herramientas open source (gratuítas) o herramientas comerciales (de pago).

- Herramientas de gestión de pruebas
- Herramientas para pruebas funcionales
- Herramientas para pruebas de carga y rendimiento.

HERRAMIENTAS OPEN SOURCE:

I) Herramientas de gestión de pruebas

- * Bugzilla Testopia
- FitNesse
- qaManager
- qaBook
- RTH (open source)
- Salome-tmf
- Squash TM
- Test Environment Toolkit
- TestLink
- Testitool
- XQual Studio
- Radi-testdir
- Data Generator

HERRAMIENTAS OPEN SOURCE:

2) Herramientas para pruebas funcionales

- Selenium
- Soapui
- Watir (Pruebas de aplicaciones web en Ruby)
- WatiN (Pruebas de aplicaciones web en .Net)
- Capedit
- Canoo WebTest
- Solex
- Imprimatur
- SAMIE
- ITP
- WET
- WebInject

HERRAMIENTAS OPEN SOURCE:

3) Herramientas para pruebas de carga y rendimiento

- FunkLoad
- FWPTT load testing
- loadUI
- jmeter

HERRAMIENTAS COMERCIALES: PARA LA VENTA

I) Herramientas de gestión de pruebas

HP Quality Center/ALM

QA Complete

qaBook

T-Plan Professional

SMARTS

QAS.Test Case Studio

PractiTest

SpiraTest

TestLog

ApTest Manager

Zephyr

HERRAMIENTAS COMERCIALES: PARA LA VENTA

2) Herramientas para pruebas funcionales

QuickTest Pro

Rational Robot

Sahi

SoapTest

Test Complete

QA Wizard

Squish

vTest

Internet Macros

HERRAMIENTAS COMERCIALES: PARA LA VENTA

3) Herramientas para pruebas de carga y rendimiento

HP LoadRunner

LoadStorm

NeoLoad

WebLOAD Professional

Forecast

ANTS – Advanced .NET Testing
System

Webserver Stress Tool

Load Impact

QUE SON LAS PRUEBAS UNITARIAS

Se hace referencia a los componentes como módulos, clases o unidades, debido a que los desarrolladores pueden estar involucrados en la ejecución de pruebas, pueden llamarse también pruebas de desarrollador.

Es importante mencionar que estas pruebas las ejecuta el desarrollador verificando que su código cumple con lo solicitado y no ha violado ningún estándar que ponga en riesgo la estabilidad del sistema.

Las pruebas de componente podrán comprobar características funcionales y no funcionales de un sistema.

PRUEBAS INTEGRALES

Las pruebas integrales también son conocidas como pruebas de interfaz, ya que comprueban la interacción entre componentes.

Las pruebas de integración asumen que los módulos ya han sido probados de manera individual (pruebas unitarias).

Implican una progresión ordenada de pruebas que van desde los componentes o módulos y que culminan en el sistema completo.

El orden de integración elegido afecta a diversos factores como los siguientes:

- La forma de preparar casos
- Las herramientas necesarias
- El orden de codificar y probar los módulos
- El costo de la depuración
- El costo de preparación de casos

ENFOQUE DE LAS PRUEBAS INTEGRALES

Ascendente. Cuando el desarrollo del sistema comienza con piezas unitarias y crece hasta formar módulos.

Descendente. Cuando se muestra el *front* de la aplicación pero está hueca por dentro y se comienza a trabajar desde los módulos más grandes hasta llegar al detalle o a las partes unitarias.

Big Bang. Que va de la mano con *xtreme programing* porque el desarrollo no tiene un orden específico y puede ser al azar.

Cualquier enfoque distinto a estos es llamado *ad-hoc* que encaja con el modelo de desarrollo tradicional V.

PRUEBAS DEL SISTEMA

Las pruebas de sistema se llevan a cabo cuando todo el desarrollo ha sido culminado y tenemos una versión preliminar del sistema que saldrá a producción. Esta etapa de pruebas consiste en probar un sistema integrado con el objeto de comprobar el cumplimiento de requisitos especificados.

Las pruebas se hacen con un enfoque desde el punto de vista del usuario.

Las pruebas de sistema se desarrollan utilizando casos de prueba funcionales y no funcionales. Las pruebas funcionales confirman que los requisitos para un uso específico previsto han sido cumplidos (validación).

Las pruebas de sistema no funcionales verifican los atributos de calidad no funcionales, lo vimos en el estándar ISO 9126, con los atributos funcionales y no funcionales de la calidad.

PRUEBAS FUNCIONALES

Tal y como su nombre lo indica las pruebas funcionales se enfocan en validar la correcta implementación de las necesidades del cliente. La funcionalidad puede ser vinculada a los datos de entrada y de salida. Los datos de entrada serán ejecutados y mostrarán un resultado y dicho resultado será comparado con el resultado esperado (comportamiento), este proceso se muestra en la figura 5.1.



Figura 5.1 Pruebas funcionales

PRUEBAS NO FUNCIONALES

Las pruebas no funcionales revisan las características implícitas del sistema, lo que las hace difícil de validar.

Se enfocan a las características de un software, existen pruebas no funcionales que miden atributos que no podrían ser validados sin el uso de una herramienta, a continuación se describen este tipo de pruebas:

Carga. Pruebas a un sistema cubriendo la demanda esperada.

14

Rendimiento. Rapidez con la cual un sistema ejecuta una determinada función.

Estrés. Someter a la aplicación a una carga mucho mayor a la esperada y evaluar la capacidad del sistema de seguir su operación después de este tipo de demanda.

Existen muchas otras como: seguridad, estabilidad, volumen, robustez.

Otros aspectos no funcionales de calidad:

Fiabilidad. El sistema debe ser confiable a los usuarios para que puedan operar sin preocuparse de sus datos o de la mala operación de un sistema.

Mantenibilidad. El código debe cumplir con ciertas normas y estándares para evitar que se encuentre de forma compleja, el código siempre debe estar comentado para cada función realizada.

Portabilidad. Se deben asegurar las características del sistema; en qué sistema operativo operará, cuál será el navegador de preferencia, etcétera.

TÉCNICAS DE PRUEBAS ESTÁTICAS

Las técnicas estáticas de pruebas comprenden métodos donde no se ejecutan los componentes u objetos de prueba (sistema).

Las pruebas estáticas incluyen: revisiones y análisis estáticos. Estas pruebas complementan los métodos dinámicos. De esta manera se pueden detectar causas de fallos en lugar de fallos.

La detección temprana de errores ahorra costos.

Ventajas

- Costos más bajos
- Los defectos en la documentación son detectados y corregidos de manera temprana
- Los documentos de alta calidad mejoran el proceso de desarrollo

Desventajas

- Pueden presentarse situaciones de tensión con el autor
- Inversión considerable de tiempo (10% - 15% del presupuesto total)

TÉCNICAS DE PRUEBAS DINÁMICAS: DE CAJA BLANCA

El objetivo es analizar el objeto de prueba, ejecutando el objeto de prueba. Se llevan a cabo por medio de revisiones y herramientas.

- Complementan los métodos estáticos
- Costos más bajos, la corrección de errores sucede en etapas tempranas.

Prevención de defectos

Las técnicas de caja blanca, examinan la parte interna del programa, siempre se está observando el código, y los casos de prueba están basados en la estructura interna del programa (figura 6.2). Por ello, la implementación de estas pruebas depende de la disponibilidad del código fuente. Este tipo de pruebas permiten generar casos para ejercitar y validar los caminos de cada módulo, las condiciones lógicas, los bucles, etcétera.

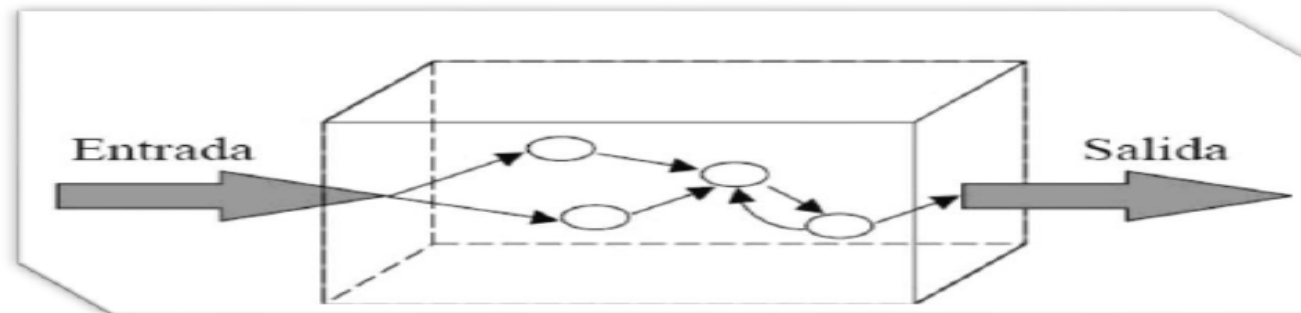


Figura 6.1 Caja blanca

TÉCNICAS DE PRUEBAS DINÁMICAS: DE CAJA NEGRA

La técnica de caja negra, se enfoca en probar el sistema sin tomar en cuenta la estructura interna del mismo, su objetivo es validar que las salidas sean las esperadas (figura 6.2).

Se centra en encontrar las circunstancias en las que el sistema no se comporta conforme a las especificaciones establecidas.

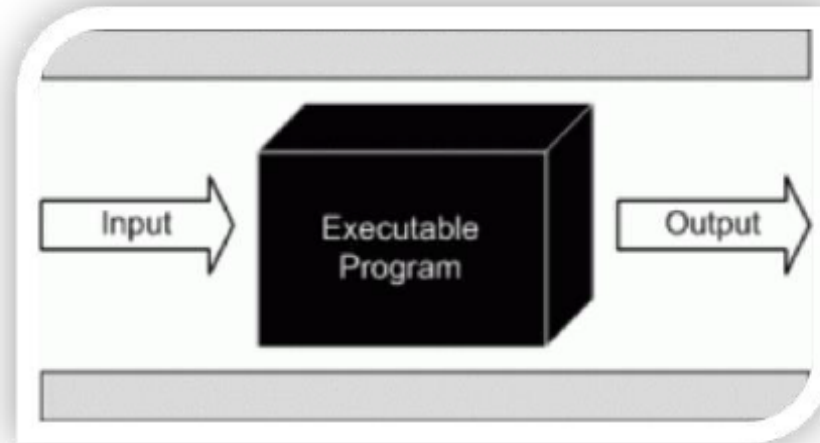


Figura 6.2 Caja negra

TÉCNICAS DE CAJA NEGRA MAS COMUNES

Las técnicas más comunes de caja negra son:

Partición de equivalencia (posibles valores divididos en clases, valores de entrada y valores de salida). Se agrupan todos los valores para los cuales se espera que el programa tenga un comportamiento común

23

(rango de valores), y esa es una clase de equivalencia, existen clases de equivalencia válidas y clases de equivalencias inválidas.

Valores límite. Complementan a la partición equivalente, se debe prestar mucha atención en que los límites deben estar correctamente definidos y programados.

Transición de estado. La transición de estados nos enuncia que todo sistema se mueve por transiciones de un paso a otro, nos podemos guiar por las transacciones válidas y las transiciones inválidas.

Tablas de decisión. Estas pruebas consideran que para encontrar el resultado esperado deben estar en conjunto varias condiciones que son los detonadores del resultado, llamado como causa y efecto.

EJEMPLO DE LA TÉCNICA DE LA CAJA NEGRA

Ejemplo:

Se tiene el siguiente formato de registro

Número De Empleado	Nombre	Meses Trabajando	Directivo
--------------------	--------	------------------	-----------

Dónde:

Número de empleado es un campo de tres dígitos enteros positivos, sin incluir el 000.

Nombre es un campo alfanumérico de máximo 30 caracteres.

Meses trabajando es un campo que indica el número de meses que lleva el empleado en la empresa. Entero positivo de 3 dígitos, incluyendo el 000.

Directivo es un campo de un solo carácter que puede ser "+" para indicar que el empleado es un directivo o "-" para indicar que no lo es.

De acuerdo a lo anterior, tenemos que:

Condición	Clases válidas	Clases inválidas
Número de empleado	-Número de tres dígitos, enteros positivos $000 > x \leq 999$	-Número menor a tres dígitos -Número mayor a tres dígitos -x = 000 -Números negativos -Números decimales -Caracteres no numéricos -Cadena nula

CONTINUA EJEMPLO

Nombre	-Cadena alfanumérica de máximo 30 caracteres	-Cadena mayor a 30 caracteres -Cadena nula
Meses trabajando	-Número de tres dígitos, enteros positivos $000 \geq x \leq 999$	-Número menor a tres dígitos -Número mayor a tres dígitos -Números negativos -Números decimales -Caracteres no numéricos -Cadena nula
Directivo	-Cadena de 1 carácter $x = +$ $x = -$	-Cadena nula -Cadena de más de 1 carácter -Carácter diferente a "+" o "-"

METODOLOGÍA APLICADA POR LOS DESARROLLADORES E INGENIEROS EN LA EJECUCIÓN DE PRUEBAS AL SW

CERTIFICACIÓN DE APLICACIÓN WEB "AGENDA"

Definición del proceso de pruebas

Contenido

- Propósito
- Contexto actual
- Objetivo
- Estrategia de pruebas
- Alcance
- Criterios de suspensión y reanudación de pruebas
- Requerimientos de prueba
- Diseño de casos de prueba
 - Pruebas funcionales
 - Pruebas en la etapa de sistema
 - Técnica de pruebas caja negra
 - Partición equivalente
 - Valores límite
 - Casos de uso
- Resumen de ejecución
- Defectos detectados
- Evaluación de riesgos
- Evaluación y cierre
- Liberación a producción
- Conclusiones

COMO SE DOCUMENTA UNA PRUEBA

c) Casos de Uso

Identificador	Nombre
CU001	Agregar contacto
Requerimiento de prueba	
REQ003	
Precondiciones	
• NA	
Objetivo	
• Permitir al usuario crear un contacto dentro de la agenda web	
Flujo Principal	
<ol style="list-style-type: none">1. El caso de uso inicia cuando el usuario ejecuta la aplicación agenda en su pc.2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos.3. El usuario ingresa los campos: nombre, apellido, correo electrónico, edad y número telefónico y da clic en "Guardar".4. Si la información es correcta el módulo manda el mensaje M-001.5. El contacto se muestra en la lista de contactos.6. Fin de caso de uso.	
Flujo Alterno CU001-1	
<ol style="list-style-type: none">1. El flujo alternativo inicia cuando el usuario ejecuta la aplicación agenda en su pc.2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos.3. El usuario ingresa el campo nombre, apellido, correo electrónico, edad y número telefónico y da clic en guardar (al ingresar el usuario deja campos en blanco o en un formato incorrecto).4. Si la información es incorrecta el módulo manda el mensaje M-002 o M-003 colocando el foco en el campo donde se presente el error.5. Fin del flujo alternativo.	
Flujo Alterno CU001-2	
<ol style="list-style-type: none">1. El flujo alternativo inicia cuando el usuario ejecuta la aplicación agenda en su pc.2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos.3. El usuario ingresa el campo nombre, apellido, correo electrónico, edad y número telefónico y da clic en el botón "Cancelar".4. El módulo limpia los campos y no guarda la información.5. Fin del flujo alternativo.	

Postcondiciones			
- El contacto se guarda en la agenda web.			
Anexos			
- Matriz de Mensajes			
ID	Tipo de Mensaje	Botón Relacionado	Mensaje
M-001	Confirmación	Guardar	Contacto guardado
M-002	Error	Guardar	Existen caracteres no válidos
M-003	Error	Guardar	Campo requerido

Identificador	Nombre
CU002	Consultar todos los contactos
Requerimiento de prueba	
REQ009	
Precondiciones	
• NA	
Objetivo	
• Permitir al usuario consultar los contactos registrados en la agenda web	
Flujo Principal	
<ol style="list-style-type: none">1. El caso de uso inicia cuando el usuario ejecuta la aplicación agenda en su pc.2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos.3. El usuario da clic en el botón "Todos los contactos".4. El módulo muestra la información de los contactos existentes con opción a "Editar" y "Eliminar".5. Fin de caso de uso.	
Flujo Alterno CU002-1	
• NA	
Postcondiciones	

COMO SE DOCUMENTA UNA PRUEBA

- Los contactos existentes se muestran
Anexos
- NA

Identificador	Nombre
CU003	Editar contacto

Requerimiento de prueba

REQ005

Precondiciones

- Debe haber contactos existentes

Objetivo

- Permitir al usuario modificar contactos

Flujo Principal

1. El caso de uso inicia cuando el usuario ejecuta la aplicación agenda en su pc.
2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos.
3. El usuario da clic en el botón "Todos los contactos".
4. El módulo muestra la información de los contactos existentes con opción a "Editar" y "Eliminar".
5. El usuario da clic en el botón "Editar" del contacto que desee modificar.
6. El módulo presenta la información del contacto habilitada para su edición.
7. El usuario realiza cambios en alguno(s) de los campos: nombre, apellido, correo electrónico, edad y/o teléfono y da clic en "Actualizar".
8. El módulo muestra mensaje M-004 y muestra el contacto con la información actualizada.
9. Fin de caso de uso.

Flujo Alterno CU003-1

1. El flujo alterno inicia cuando el usuario ejecuta la aplicación agenda en su pc.
2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos.
3. El usuario da clic en el botón "Todos los contactos".
4. El módulo muestra la información de los contactos existentes con opción a "Editar" y "Eliminar".
5. El usuario da clic en el botón "Editar" del contacto que desee modificar.
6. El módulo presenta la información del contacto habilitada para su edición.
7. El usuario realiza cambios en alguno(s) de los campos: nombre, apellido, correo electrónico, edad y/o número telefónico y da clic en "Cancelar".
8. El módulo no guarda los cambios.
9. El módulo muestra pantalla todos los contactos.

10. Fin del flujo alterno.

Postcondiciones

- El contacto se actualiza

Anexos

- Matriz de Mensajes

Id	Tipo de Mensaje	Botón Relacionado	Mensaje
M-004	Confirmación	Actualizar	Contacto actualizado

Identificador	Nombre
CU004	Eliminar contacto

Requerimiento de prueba

REQ007

Precondiciones

- Debe haber contactos existentes

Objetivo

- Permitir al usuario borrar contactos

Flujo Principal

1. El caso de uso inicia cuando el usuario ejecuta la aplicación agenda en su pc.
2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos.
3. El usuario da clic en el botón "Todos los contactos".
4. El módulo muestra la información de los contactos existentes con opción a "Editar" y "Eliminar".
5. El usuario da clic en el botón "Eliminar" del contacto que desee borrar.
6. El módulo muestra mensaje M-005.
7. El usuario confirma el mensaje.
8. El módulo muestra mensaje M-006.
9. Si el contacto fue eliminado de manera correcta ya no se muestra en la lista.
10. Fin de caso de uso.

Flujo Alterno CU004-1

CONTINUA

1. El flujo alterno inicia cuando el usuario ejecuta la aplicación agenda en su pc.
2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos.
3. El usuario da clic en el botón "Todos los contactos".
4. El módulo muestra la información de los contactos existentes con opción a "Editar" y "Eliminar".
5. El usuario da clic en el botón "Eliminar" del contacto que desee borrar.
6. El módulo muestra mensaje M-005.
7. El usuario da clic en "Cancelar".
8. El módulo no elimina el contacto.
9. Fin del flujo alterno.

Postcondiciones

- El contacto es eliminado

Anexos

- Matriz de Mensajes

ID	Tipo de Mensaje	Botón Relacionado	Mensaje
M-005	Confirmación	Eliminar	¿Desea eliminar contacto?
M-006	Confirmación	Aceptar	Contacto eliminado

Identificador	Nombre
CU005	Buscar Contacto
Requerimiento de prueba	
REQ008	
Precondiciones	
<ul style="list-style-type: none"> • NA 	
Objetivo	
<ul style="list-style-type: none"> • Permitir al usuario realizar búsqueda de contactos 	
Flujo Principal	
<ol style="list-style-type: none"> 1. El caso de uso inicia cuando el usuario ejecuta la aplicación agenda en su pc. 2. El módulo presenta formulario para crear nuevo contacto, campo para realizar búsqueda y opción para consulta de todos los contactos. 3. El usuario ingresa el texto a buscar en el campo "Buscar contactos" y da clic en el botón "Buscar". 	

4. El módulo muestra los contactos que coinciden con el texto ingresado.
5. En caso de no existir coincidencias el módulo no muestra información.
6. El usuario Fin de caso de uso.

Flujo Alterno CU004-1

- NA

Postcondiciones

- Se muestran resultados de la búsqueda en caso de existir.

Anexos

- NA

9.9 Resumen de ejecución

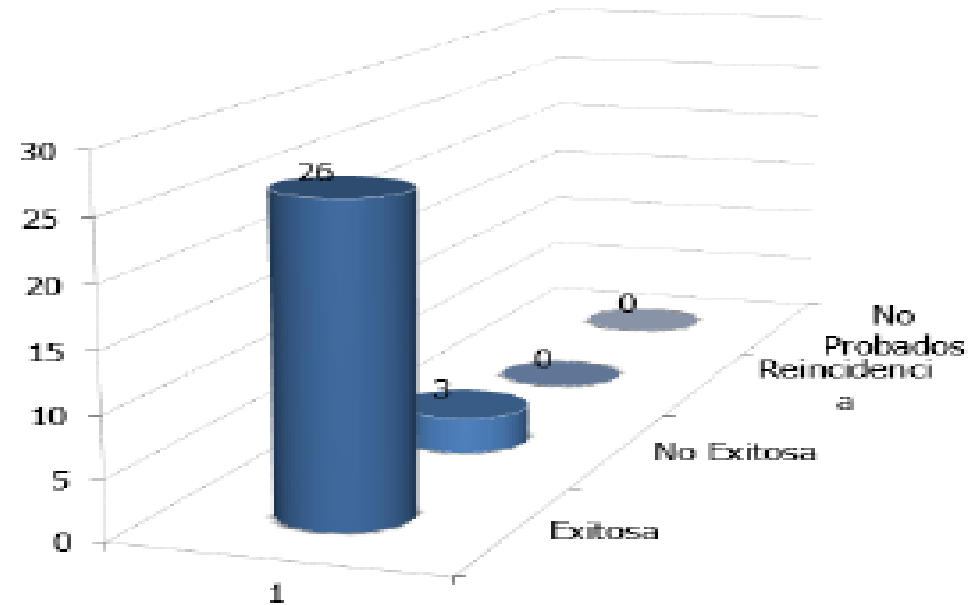
ID	Requerimiento	ID	Tipo	Caso de prueba	Resultado
3	Agregar contacto	50	TTF	ClaseInválida.NúmerosNegativos.Teléfono	Pass
		26	TTF	CamposRequeridos	Pass
		16	TTF	ClaseInválida.CaracteresEspecialesYNuméricos.Apellido	Pass
		14	TTF	ClaseInválida.CaracteresEspecialesYNuméricos.Nombre	Pass
		22	TTF	ClaseInválida.CaracteresNoNuméricos.Edad	Pass
		21	TTF	ClaseInválida.CaracteresNoNuméricos.Teléfono	Pass
		18	TTF	ClaseInválida.CorreoElectrónico	Pass
		23	TTF	ClaseInválida.NúmerosDecimales.Edad	Pass
		49	TTF	ClaseInválida.NúmerosDecimales.Teléfono	Pass
		19	TTF	ClaseInválida.X<10Dígitos.Teléfono	Pass
		20	TTF	ClaseInválida.X>10Dígitos.Teléfono	Pass
		17	TTF	ClaseInválida.X>15Caracteres.Apellido	Pass
		15	TTF	ClaseInválida.X>15Caracteres.Nombre	Pass
		44	TTF	ContactoDuplicado	Pass
		48	TTF	ValidarCorreoElectrónico	Pass
		25	TTF	ValoresLimiteInválidos.Edad	Pass
		28	TTF	Cancelar	Pass
		13	TTF	ClasesVálidas	Pass
		24	TTF	ValoresLimiteVálidos.Edad	Pass
		39	TTF	ClaseInválida.X>15Caracteres.CorreoElectrónico	Pass
5	Editar contacto	27	TTF	EditarContacto	Pass
		29	TTF	EditarContacto - Cancelar	Pass
7	Eliminar contacto	30	TTF	EliminarContacto	Pass
		31	TTF	EliminarContacto - Cancelar	Pass
8	Buscar contacto	33	TTF	BuscarContactoNoExistente	Pass
		32	TTF	BuscarContactoExistente	Pass
9	Consultar todos los contactos	34	TTF	ConsultarContactos	Pass
42	Look & feel	36	TTF	L&F - PáginaPrincipal	Fail
		37	TTF	L&F - TodosLosContactos	Fail

Estatus de prueba

Ejecución

Estatus	Ciclo 1
Exitosa	26
No Exitosa	3
Reincidencia	0
No Probados	0

General	
Requerimientos de prueba	6
Casos de Prueba	29
Happy Path	1
Automatizados	0



9.10 Defectos detectados

ID	Defecto	Riesgo	Consecuencia	Impacto	Estatus
45	Homogenizar etiquetas	Ninguno	NA	NA	Cerrado
46	Cabeceras	Ninguno	NA	NA	Cerrado
47	Etiqueta buscar contacto	Ninguno	NA	NA	Cerrado

WEBLIOGRAFÍA

- Guía de certificación del ISTQB® nivel básico

- <http://standards.ieee.org/>

10.01.14

- <http://www.fiuxy.com/noticias/1667647-20-errores-informaticos-excelente.html>

27.01.14

- Ingeniería del software.

Séptima edición. Ian Sommerville

Pearson educación. S. A., Madrid 2005

- Ingeniería del software. Un enfoque práctico.

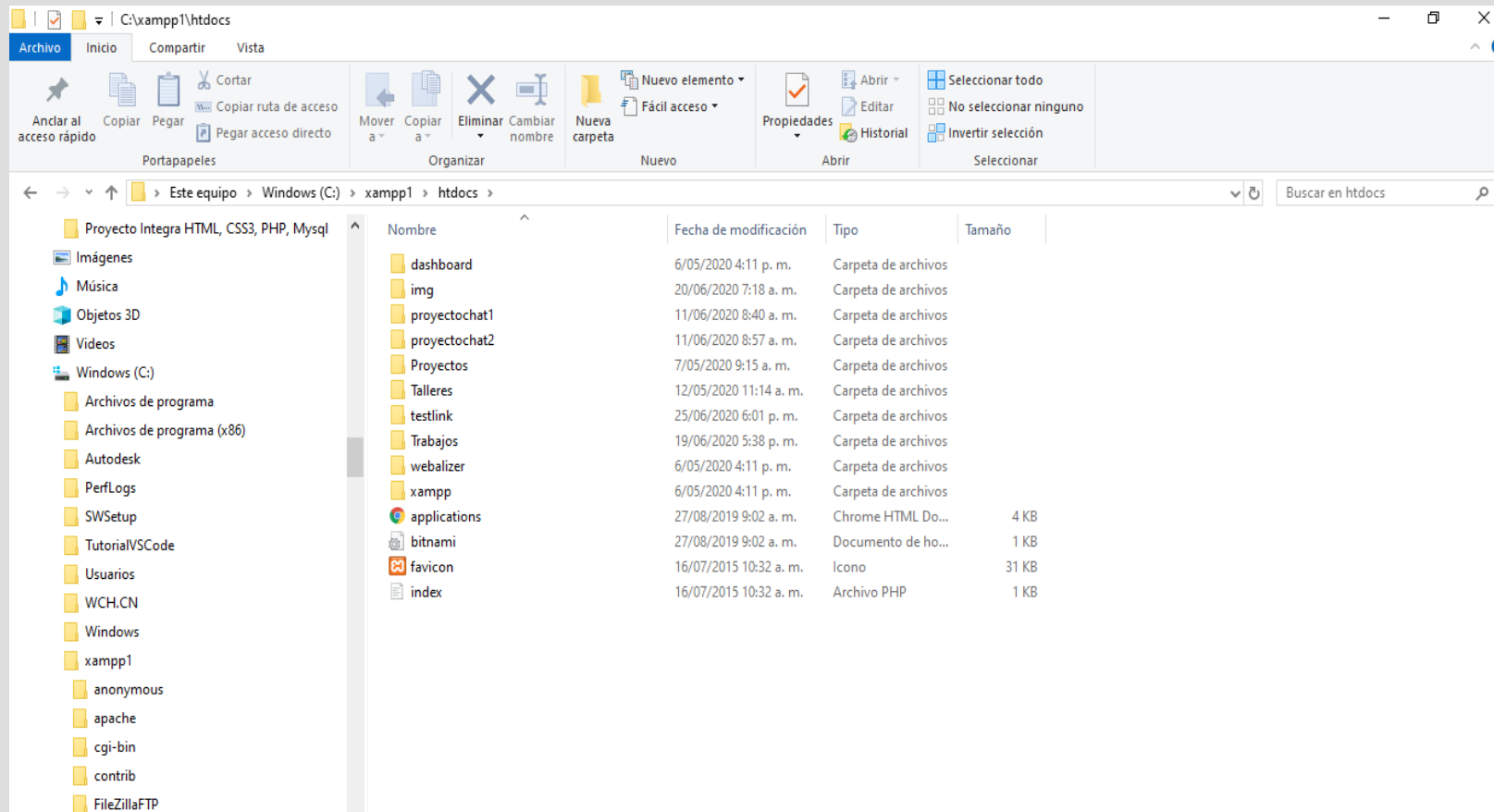
Séptima edición. Roger S. Pressman, Ph. D.

McGraw Hill, 2010

LABORATORIO DE CLASE: INTALACION DEL SOFTWARE DE PRUEBAS: TESTLINK

1. Pagina para descargar software: <https://sourceforge.net/projects/testlink/>
- 2.Video: <https://www.youtube.com/watch?v=D6jBjvkNIDM&t=9s>
- 3.Verificar el video a partir del minuto 5 ya que el proceso de instalación del Xampp ya es conocido por todos
4. El SW de pruebas se coloca una versión en la plataforma Territorium

COLOCAMOS LA CARPETA TESTLINK EN HTDOCS DE XAMPP



HTTP://LOCALHOST/TESTLINK/INSTALL/INDEX.PHP

Warning: Creating default object from empty value in C:\xampp1\htdocs\testlink\cfg\const.inc.php on line 370

TestLink Development Prague 1.9.6 (Direct Links) Installation

You are installing TestLink Development Prague 1.9.6 (Direct Links)

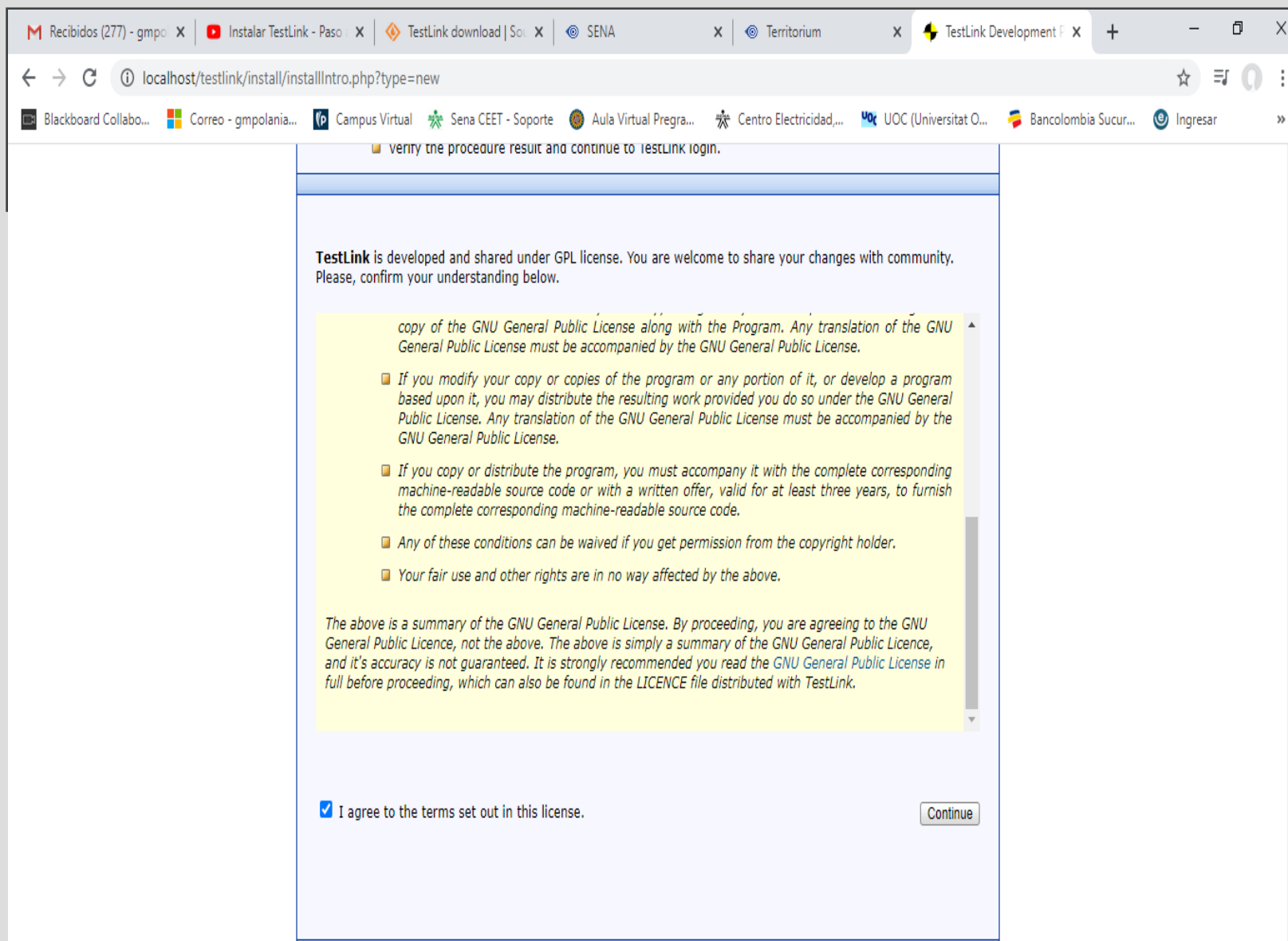
Migration from 1.9.3/4/5 to Development Prague 1.9.6 (Direct Links) require Database changes that has to be done MANUALLY.

For information about Migration from older version please read README file provided with installation.

Please read Section on README file or go to www.teamst.org (Forum: TestLink 1.9.4 and greater News,changes, etc)

Open [Installation manual](#) for more information or troubleshooting. You could also look at [README](#) or [Changes Log](#). You are welcome to visit our [forum](#) to browse or discuss.

[New installation](#)



DEBEMOS CORREGIR LOS PATH DE LOS DOS ERRORES (FAILED!)

estLink - Paso - x | TestLink download | So... x | SENA x | Territorium x | TestLink Develo

ll/installCheck.php?licenseOK=on

nia... Campus Virtual Sena CEET - Soporte Aula Virtual Pregra... Centro Electricidad,... UOC (Universitat O...

Checking JSON library

authentication will works).
OK

Read/write permissions

For security reasons we suggest that directories tagged with [S] on following messages, will be made UNREACHABLE from browser.
Give a look to README file, section 'Installation & SECURITY' to understand how to change the defaults.

Checking if C:\xampp1\htdocs\testlink\gui\templates_c directory exists OK

Checking if C:\xampp1\htdocs\testlink\gui\templates_c directory is writable (by user used to run webserver process) OK

Checking if /var/testlink/logs/ directory exists [S] Failed!

Checking if /var/testlink/upload_area/ directory exists [S] Failed!

Unfortunately, TestLink scripted setup cannot continue at the moment, due to the above 2 errors.
Please correct the errors, and try again (reload page). If you need help figuring out how to fix the problems, please read Installation manual and visit [TestLink Forums \[click here\]](#).

VAMOS A LA CARPETA TESTLINK Y BUSCAMOS EL ARCHIVO CONFIG.INC LO ABRIMOS EN EL BLOC DE NOTAS

The screenshot shows a Windows File Explorer window with the address bar set to `C:\xampp\htdocs\testlink`. The left sidebar shows the navigation pane with the 'testlink' folder selected. The main pane displays a list of files and folders. The file `config.inc.php` is selected, and a context menu is open, showing options to open the file with various applications. The 'Bloc de notas' (Notepad) application is highlighted.

Nombre	Fecha de modificación	Tipo	Tamaño
cfg	25/06/2020 5:58 p. m.	Carpeta de archivos	
custom	25/06/2020 5:58 p. m.	Carpeta de archivos	
docs	25/06/2020 5:58 p. m.	Carpeta de archivos	
gui	25/06/2020 5:59 p. m.	Carpeta de archivos	
install	25/06/2020 5:59 p. m.	Carpeta de archivos	
lib	25/06/2020 5:59 p. m.	Carpeta de archivos	
locale	25/06/2020 5:59 p. m.	Carpeta de archivos	
logs	25/06/2020 5:59 p. m.	Carpeta de archivos	
third_party	25/06/2020 6:01 p. m.	Carpeta de archivos	
upload_area	25/06/2020 6:01 p. m.	Carpeta de archivos	
CHANGELOG	9/03/2013 12:20 p. m.	Archivo	230 KB
CODE_REUSE	9/03/2013 12:20 p. m.	Archivo	1 KB
config.inc.php	9/03/2013 12:20 p. m.	Archivo PHP	60 KB
custom_config.inc.php.example	9/03/2013 12:20 p. m.	Archivo EXAMPLE	10 KB
error.php	9/03/2013 12:20 p. m.	Archivo PHP	2 KB
firstLogin.php	9/03/2013 12:20 p. m.	Archivo PHP	4 KB
index.php	25/06/2020 5:43 p. m.	Archivo PHP	3 KB
LICENSE	9/03/2013 12:20 p. m.	Archivo	18 KB
linkto.php	9/03/2013 12:20 p. m.	Archivo PHP	10 KB
lnl.php	9/03/2013 12:20 p. m.	Archivo PHP	5 KB
login.php	9/03/2013 12:20 p. m.	Archivo PHP	8 KB
logout.php	9/03/2013 12:20 p. m.	Archivo PHP	1 KB
lostPassword.php	9/03/2013 12:20 p. m.	Archivo PHP	2 KB
README	9/03/2013 12:20 p. m.	Archivo	12 KB

¿Cómo quieres abrir este archivo?

- Visual Studio Code
- Buscar una aplicación en Microsoft Store
- Access 2016
- Adobe Acrobat Reader DC
- Blackboard Collaborate Launcher
- Bloc de notas**
- Firefox
- ☒ Siempre usar esta aplicación para abrir archivos .php

Aceptar

SE VERA ASI: Y BUSCAMOS LA LINEA DEL PATH QUE INDICA EL ERROR EN EL ARCHIVO PARA REEMPLAZARLO

config.inc.php: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
<?php
/**
 * TestLink Open Source Project - http://testlink.sourceforge.net/
 * This script is distributed under the GNU General Public License 2 or later.
 *
 * @filesource config.inc.php
 * @package TestLink
 * @copyright 2005-2011, TestLink community
 * @link http://www.teamst.org/index.php
 */
```

Buscar

Buscar: /var/testlink/logs/

Dirección: ☐ Subir ☒ Bajar

☐ Coincidir mayúsculas y minúsculas ☐ Ajuste automático

Cancelar

Buscar siguiente

config.inc.php

Buscar

Buscar: /var/testlink/logs/

Dirección: ☐ Subir ☒ Bajar

☐ Coincidir mayúsculas y minúsculas ☐ Ajuste automático

Cancelar

Buscar siguiente

config.inc.php

Recibidos (277) - gmpo x Instalar TestLink - Paso x TestLink download | So x SENA x Territorium x TestLink Development F x

localhost/testlink/install/installCheck.php?licenseOK=yes

config.inc.php: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
/** @var string Default level of logging (NONE, ERROR, INFO, DEBUG, EXTENDED)
 * is not used by tlLogger, we need to change this in future.
 */
$tlcfg->log_level = 'INFO';

/** @var string Path to store logs -
 * for security reasons (see http://itsecuritysolutions.org/2012-08-13-
 * put it out of reach via web or configure access denied.
 */
$tlcfg->log_path = '/var/testlink/logs/'; /* unix example */
```

Buscar

Buscar: /var/testlink/logs/

Dirección: ☐ Subir ☒ Bajar

☐ Coincidir mayúsculas y minúsculas ☐ Ajuste automático

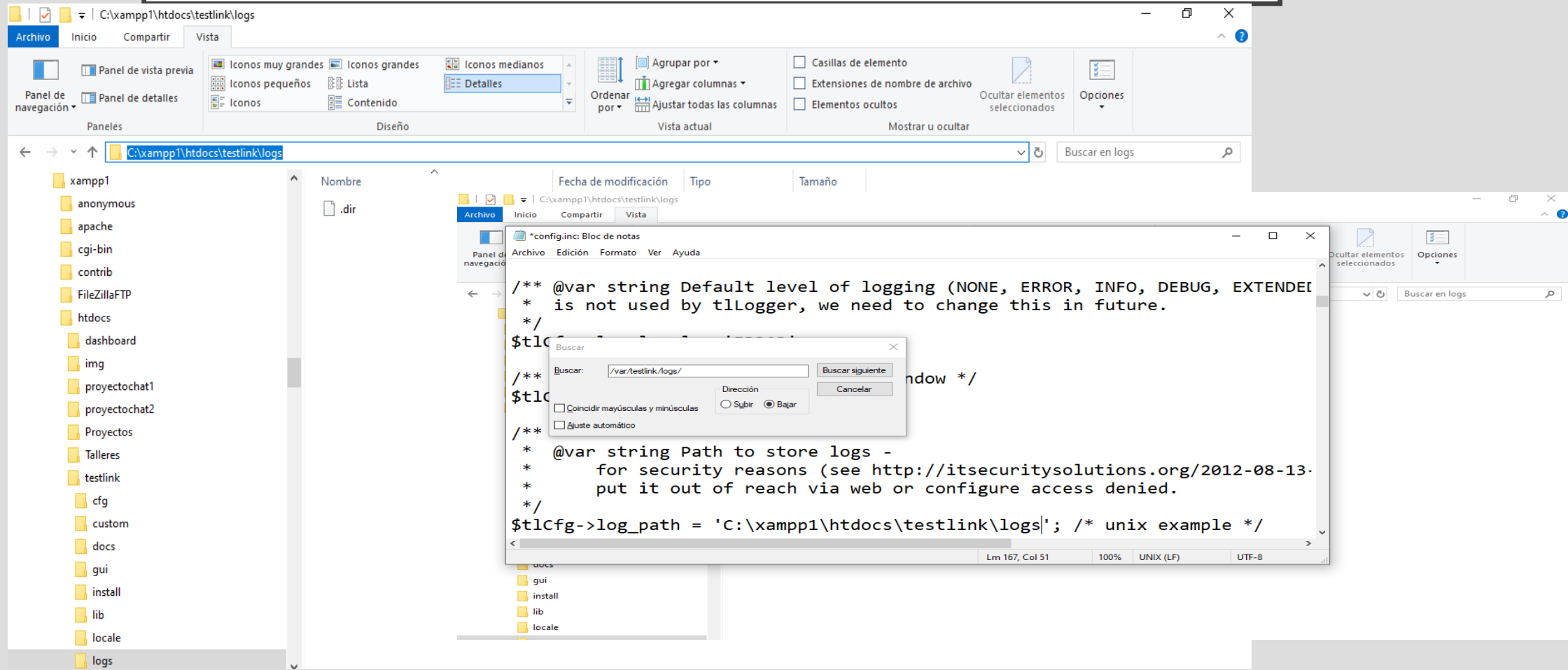
Cancelar

Buscar siguiente

config.inc.php

Unfortunatamente, TestLink scripted setup cannot continue at the moment, due to the above 2 errors. Please correct the errors, and try again (reload page). If you need help figuring out how to fix the problems, please read Installation manual and visit [TestLink Forums](#) [click here].

REEPLAZAMOS POR:

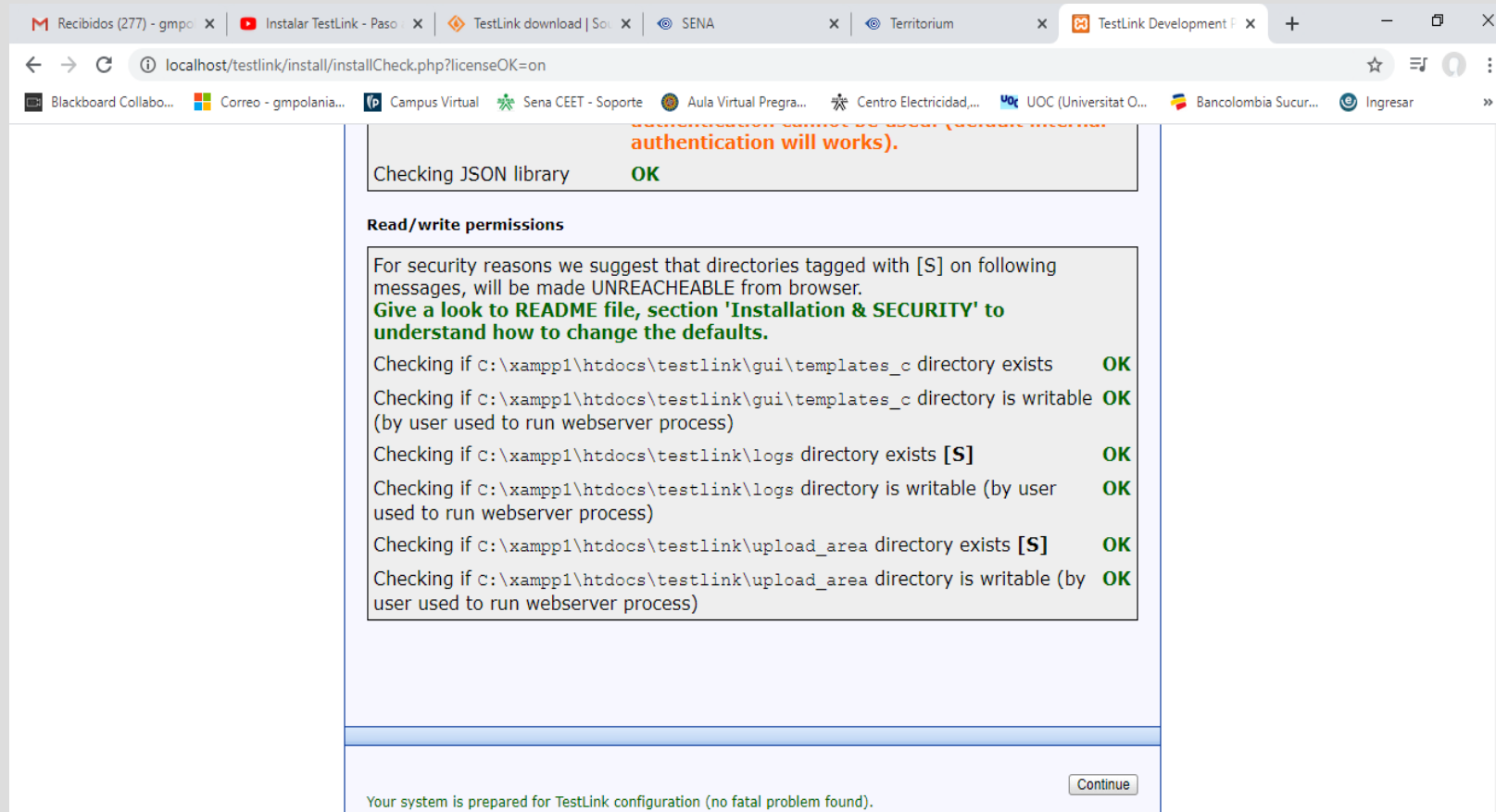


EN EL MISMO DOCUMENTO BUSCAMOS EL PATH DEL OTRO ARCHIVO: /VAR/TESTLINK/UPLOAD_AREA/ Y SE REEMPLAZA POR LA UBICACIÓN EN EL DISCO DURO

The screenshot displays a Windows desktop environment. At the top, a web browser window shows multiple tabs, including 'Recibidos (277) - gmpc', 'Instalar TestLink - Paso', 'TestLink download | So', 'SENA', 'Territorium', and 'TestLink Development'. Below the browser, a text editor window titled '*config.inc: Bloc de notas' is open, showing configuration code for TestLink. The code includes comments about repository types and a line defining the repository path: `$g_repositoryPath = '/var/testlink/upload_area/'; /* unix example */`. A search dialog box is overlaid on the text editor, with the search term `/var/testlink/upload_area/` entered. Below the text editor, a file explorer window is open, showing the directory `C:\xampp1\htdocs\testlink\upload_area`. The file explorer displays a list of files and folders, including `apache`, `cgi-bin`, `contrib`, `FileZillaFTP`, and `htdocs`. The file explorer's right pane shows a table of files and folders:

Nombre	Fecha de modificación	Tipo	Tamaño
.dir	9/03/2013 12:20 p. m.	Archivo DIR	1 KB
.htaccess	9/03/2013 12:20 p. m.	Archivo HTACCESS	1 KB

VOLVEMOS AL NAVEGADOR Y DEBE QUEDAR YA CORREGIDOS LOS 2 ERRORES Y CONTINUAMOS LA INSTALACION



SALE UN FORMULARIO Y MODIFICAMOS UNOS VALORES: ROOT Y ADMIN-ADMIN

TestLink download | Sena

TestLink Development Program 3.9.6 (Direct Link) - New Installation

- Acceptance of license
- Verification of system and configuration requirements
- Definition of DB access**
- Create DB, testlink DB user, structures and default data & create configuration file.
- Verify the procedure result and continue to TestLink login.

Database Configuration

Define your database to store TestLink data:

Database Type

Database host

Note: In the case that you DB connection don't use **STANDARD PORT** for , you need to add 'port_number', at the end Database host parameter. Example: you use MySQL running on port 6606, on server matrix then Database host will be matrix:6606

Enter the name of the TestLink database. The installer will attempt to create it if not exists.

Database name

Disallowed characters in Database Name:
The database name can contain any character that is allowed in a directory name, except '?', '\', or '.'.

Table prefix (optional)

Note: This parameter should be empty for the most of cases.
Using a Database shared with other applications: Testlink can be installed (using this installer) on a existing database used by another application, using a table prefix.
Warning! PART OF INSTALLATION PROCESS CONSISTS on dropping all TestLink tables present on the database/schema (if any TestLink table exists). Backup your Database Before installing and load after this process.

Set an existing database user with administrative rights (root):

Database admin login

Database admin password

This user requires permission to create databases and users on the Database Server.
These values are used only for this installation procedures, and is not saved.

Define database User for Testlink access:

TestLink DB login

TestLink DB password

This user will have permission only to work on TestLink database and will be stored in TestLink configuration.
All TestLink requests to the Database will be done with this user.

TestLink - Paso | TestLink download | Sena

stall/installDbInput.php?

Note: This parameter should be empty for the most of cases.
Using a Database shared with other applications: Testlink can be installed (using this installer) on a existing database used by another application, using a table prefix.
Warning! PART OF INSTALLATION PROCESS CONSISTS on dropping all TestLink tables present on the database/schema (if any TestLink table exists). Backup your Database Before installing and load after this process.

Set an existing database user with administrative rights (root):

Database admin login

Database admin password

This user requires permission to create databases and users on the Database Server.
These values are used only for this installation procedures, and is not saved.

Define database User for Testlink access:

TestLink DB login

TestLink DB password

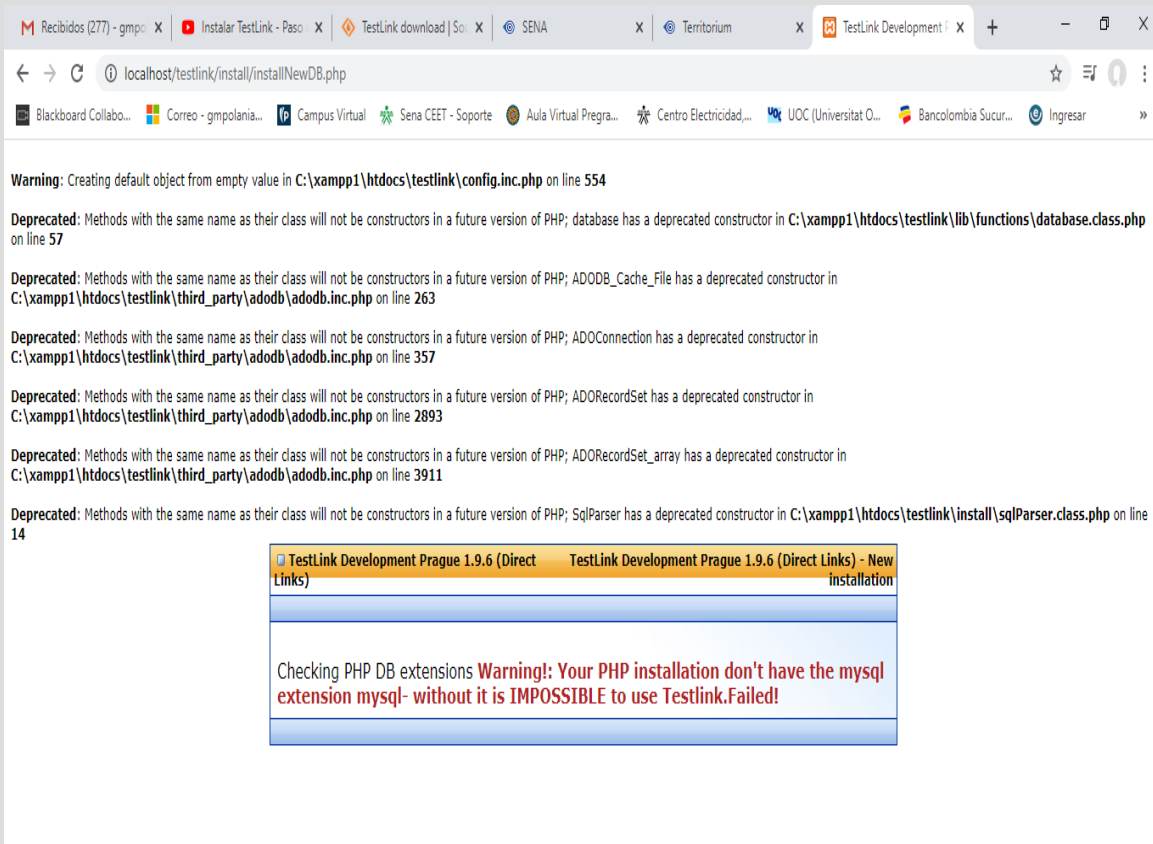
This user will have permission only to work on TestLink database and will be stored in TestLink configuration.
All TestLink requests to the Database will be done with this user.

After successfull installation You will have the following login for TestLink Administrator:
login name: admin
password : admin

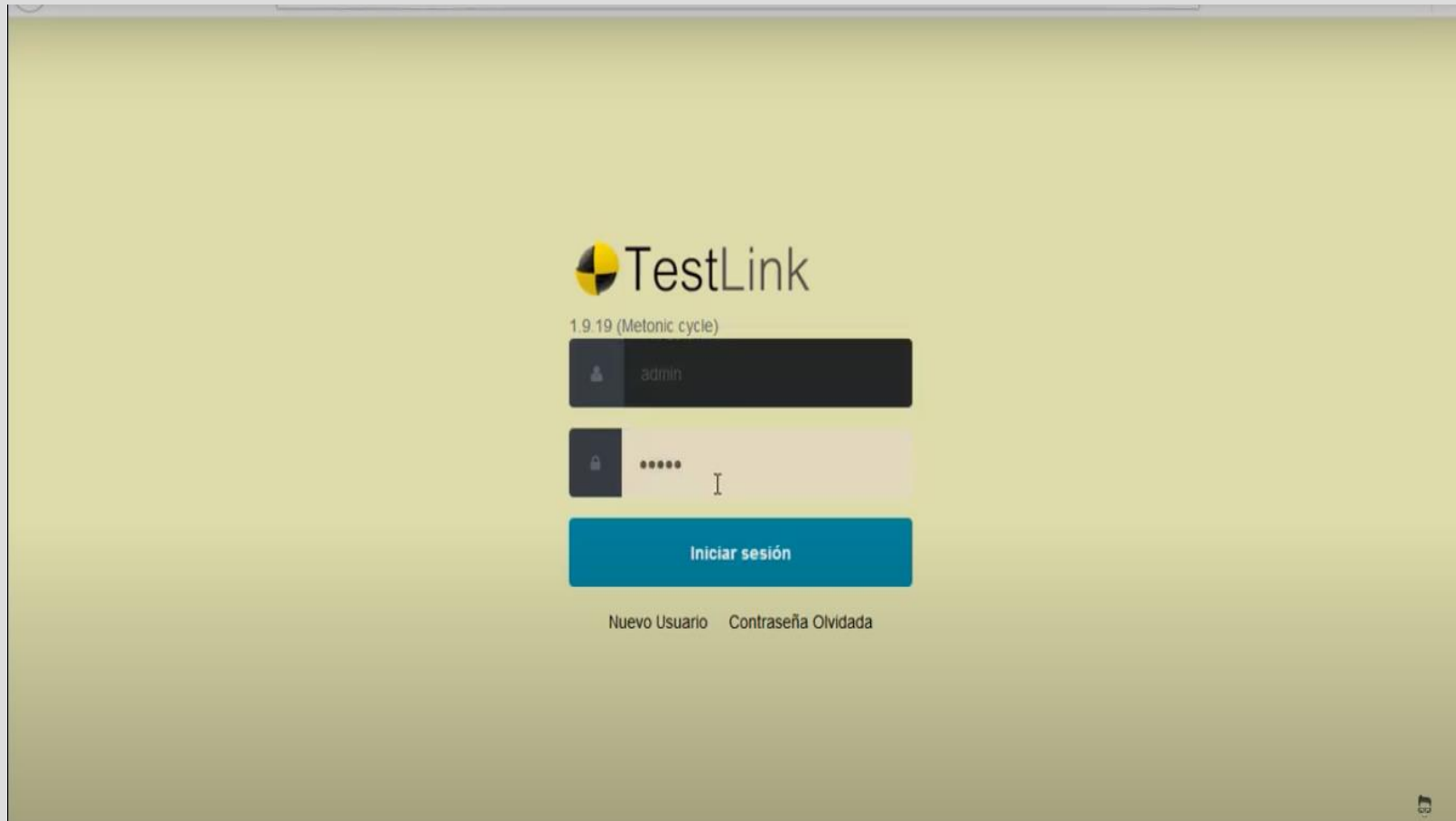
After successfull installation You will have the following login for TestLink Administrator:
login name: admin
password : admin

Process TestLink Setup!

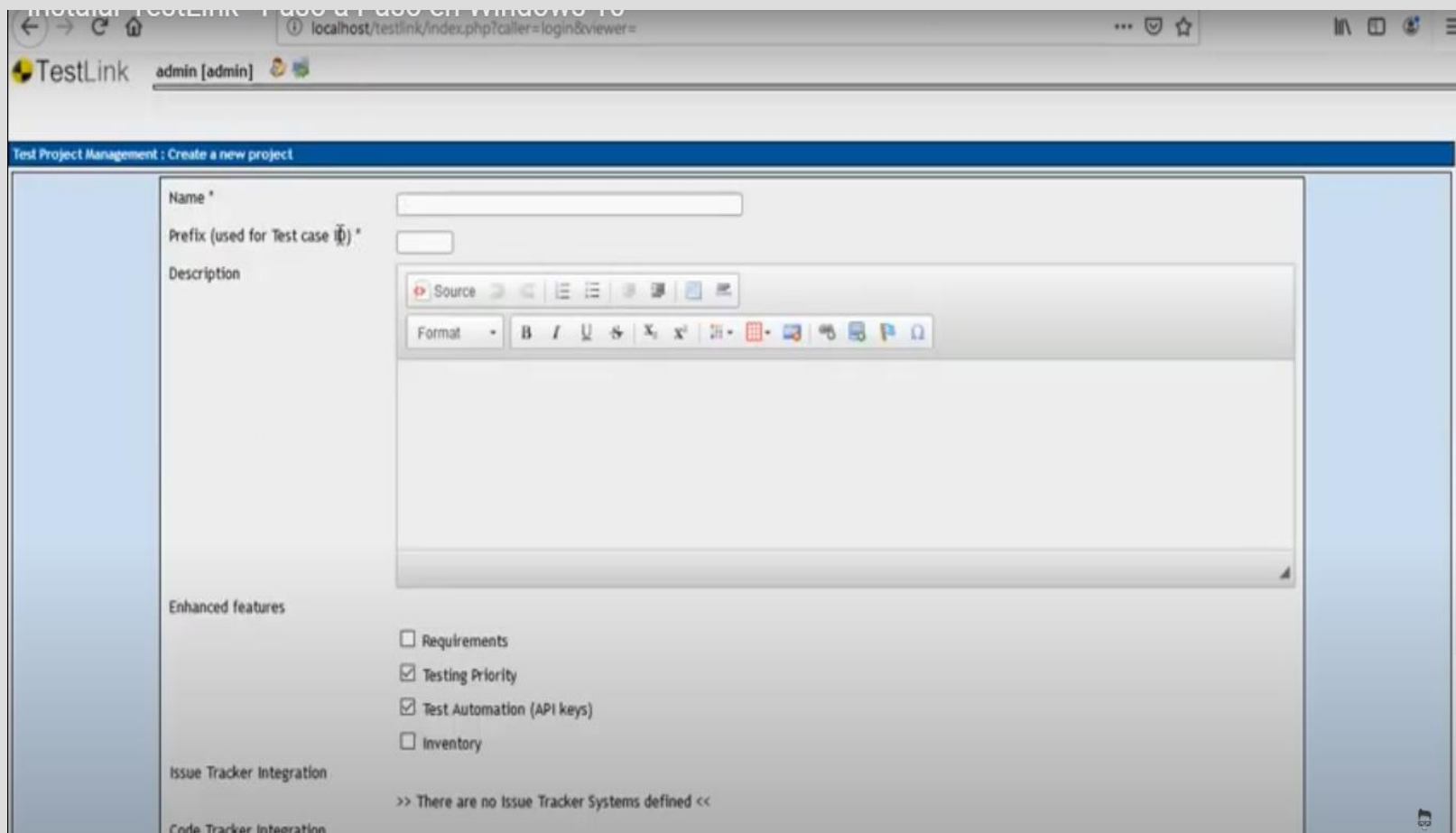
SI LA PANTALLA QUEDA CON ERRORES DEBEMOS VERIFICAR EL PATH DE LOCALHOST



DEBEN CARGAR EL PROGRAMA Y
ENTRAR CON LA CLAVE ADMIN-ADMIN



CREAN UN NUEVO PROYECTO Y SE REALIZAN LAS PRUEBAS CONFORME A LOS PLANES



The screenshot shows the TestLink web interface for creating a new project. The browser address bar indicates the URL is localhost/testlink/index.php?caller=login&viewer=. The user is logged in as 'admin [admin]'. The page title is 'Test Project Management : Create a new project'.

The form contains the following fields and options:

- Name ***: A text input field.
- Prefix (used for Test case ID) ***: A text input field.
- Description**: A rich text editor with a toolbar including 'Source', 'Format', bold, italic, underline, link, unlink, list, and other standard text formatting icons.
- Enhanced features**: A list of checkboxes:
 - ☐ Requirements
 - ☒ Testing Priority
 - ☒ Test Automation (API keys)
 - ☐ Inventory
- Issue Tracker Integration**: A section with the text '>> There are no Issue Tracker Systems defined <<'. Below this is a 'Code Tracker Integration' label.

TALLER

- 1, Bajar SW Libre de Pruebas
2. Instalar
- 3, Manual de ese sw
- 4, Investigar para que sirve y como funciona
- 5, Video de la prueba
- 6, Construir un paso a paso desde bajar el SW hasta la realización de una prueba
- 7, Explicarlo y compartirlo con los compañeros