**PROYECTO DE PRUEBA DE SOFTWARE**

**“Desarrollo de una aplicación móvil para Gestión de Notas para el Colegio “MONITOR HUASCAR”**

**FASE I:**

**ANALISIS DE REQUISITOS DE PRUEBA**

**ACTIVIDADES**

1. **Propósito del Proyecto**
2. **Alcance del Proyecto**
3. **Objetivos del Proyecto**
4. **Integrantes del Proyecto**
5. **Identificar los tipos de pruebas que debemos realizar.**
6. **Recopilar informacion sobre las prioridades y el enfoque de las pruebas.**
7. **Preparar todos los Requisitos – Matriz de trazabilidad (RTM)**
8. **Identificar los detalles del entorno de prueba de la ubicación donde debemos realizar las pruebas.**
9. **Análisis de viabilidad de automatización (¿Por qué?)**
10. **PROPOSITO DEL PROYECTO DEL PLAN DE PRUEBA**

**El propósito principal de nuestro plan de pruebas es asegurar la eficacia, confiabilidad y seguridad de la aplicación web de control de notas para el colegio Monitor Huascar. Nos enfocamos en identificar y corregir defectos potenciales, validando las funcionalidades clave para ofrecer una experiencia de usuario fluida y satisfactoria. El objetivo es garantizar que la aplicación cumpla con los estándares de calidad esperados, proporcionando a los usuarios una herramienta confiable y segura para gestionar las notas de los estudiantes de manera eficiente y sin problemas.**

1. **ALCANCE DEL PROYECTO DEL PLAN DE PRUEBAS**

**El alcance de nuestro plan de pruebas engloba todas las áreas cruciales de la aplicación web, desde la administración de usuarios y la entrada de datos hasta el procesamiento y exhibición de calificaciones. Esto abarcará la validación de la seguridad, compatibilidad con navegadores y la usabilidad general. Además, se dará especial énfasis a la integración sin contratiempos con los sistemas ya existentes en el colegio Monitor Huascar, garantizando una experiencia de uso integrada y coherente para todos los usuarios.**

1. **OBJETIVOS DEL PROYECTO DE PLAN DE PRUEBAS**

* **Verificar la Funcionalidad:**
  + **Validar que todas las funciones de la aplicación web, como el ingreso de calificaciones, el cálculo de promedios, la generación de reportes, y la gestión de usuarios, funcionen correctamente según los requisitos especificados.**
  + **Realizar pruebas exhaustivas de cada funcionalidad para identificar posibles errores, fallos o comportamientos inesperados.**
  + **Utilizar técnicas de pruebas funcionales como pruebas de caja negra y caja blanca para garantizar la cobertura completa de las funciones y escenarios de uso.**
* **Evaluar la Seguridad de la Información:**
  + **Realizar pruebas de seguridad para identificar vulnerabilidades como brechas en la autenticación, acceso no autorizado, exposición de datos sensibles, y posibles ataques como inyección de SQL o cross-site scripting (XSS).**
  + **Verificar que se implementen medidas de seguridad adecuadas, como cifrado de datos, políticas de acceso basadas en roles, auditorías de seguridad, y protección contra ataques conocidos.**
  + **Colaborar con expertos en seguridad informática para evaluar y mejorar la postura de seguridad de la aplicación.**
* **Optimizar el Rendimiento:**
  + **Realizar pruebas de carga para evaluar el rendimiento de la aplicación bajo diferentes niveles de tráfico y carga de usuarios simulados.**
  + **Identificar cuellos de botella, tiempos de respuesta lentos, y posibles problemas de escalabilidad que puedan afectar el rendimiento de la aplicación.**
  + **Implementar ajustes y optimizaciones en la infraestructura, el código fuente y la configuración para mejorar el rendimiento y la capacidad de respuesta de la aplicación.**

1. **INTEGRANTES DEL PROYECTO**

|  |  |  |  |
| --- | --- | --- | --- |
| **NRO.** | **APELLIDO** | **NOMBRE** | **CODIGO** |
| **1** | **CHAVEZ RAMOS** | **MIGUEL ALFONZO** | **PT42229274** |
| **2** | **SOTO OBREGON** | **LEYDY JANET** | **SM76355133** |
| **3** | **VALDIVIA ORTIZ DE ZEVALLOS** | **RONALD AUGUSTO** | **A19102366** |
| **4** | **VASQUES CCACCASTO** | **WALTER** | **PT71138040** |
| **5** | **VEGA ORTIZ** | **DAVID KENSHIN** | **A19202278** |

1. **TIPOS DE PRUEBAS A REALIZAR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TIPOS DE PRUEBAS** | **PRUEBAS** | **DESCRIPCION** | **EJECUCION** | **EJECUCION DESDE UN PLAN DE PRUEBA** |
| **Funcional** | **Unitaria** | **Verificación de unidades individuales del codigo fuente.** | **NODE JS / JUNIT** | **Plan/Caso de Prueba** |
| **Humo** | **Verificación inicial para asegurar la estabilidad básica.** | **NODE JS** | **Plan/Caso de Prueba** |
| **Regresión** | **Se realiza para asegurar que las nuevas modificaciones en el codigo no afecten negativamente a las funciones existentes del software** |  | **Plan/Caso de Prueba** |
| **Integración** | **Se centran en verificar que los diferentes componentes de un sistema se integren correctamente y trabajen juntos como se espere.** |  | **Plan/Caso de Prueba** |
|  | **Sanidad** | **Esto implica la revisión de buenas prácticas de codificación, identificación de posibles problemas de mantenibilidad y aseguramiento de que el codigo sea comprensible.** |  | **Plan/Caso de Prueba** |
| **No Funcional** | **Resistencia** | **Evaluación de la capacidad de respuesta frente a carga.** |  | **Plan/Caso de Prueba** |
| **Carga** | **Evaluación de la capacidad de respuesta frente a carga.** |  | **Plan/Caso de Prueba** |
| **STRESS** | **Evaluar el comportamiento del sistema bajo condiciones extremas.** |  | **Plan/Caso de Prueba** |

1. **PRIORIDADES Y ENFOQUE DE LAS PRUEBAS.**

|  |  |  |  |
| --- | --- | --- | --- |
| **TIPO DE PRUEBA** | **PRUEBAS** | **PRIORIDADES** | **COMPLEJIDAD** |
| **Funcional** | **Unitaria** | **Alta** | **Baja** |
| **Humo** | **Media** | **Baja** |
| **Regresión** | **Media** | **Alta** |
| **Integración** | **Media** | **Alta** |
| **Sanidad** | **Baja** | **Media** |
| **No Funcional** | **Resistencia** | **Media** | **Media** |
| **Carga** | **Media** | **Media** |
| **STRESS** | **Baja** | **Media** |

1. **MATRIZ DE TRAZABILIDAD DE REQUISITOS / REQUERIMIENTOS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **REQUISITO** | **TIPO** | **PRIORIDAD** | **ESTADO** | **OBJETIVO** | **FUNCIONALIDAD** | **ESTADO** |
| **001** | **El sistema deberá permitir la creación y gestión de perfiles de usuarios para profesores y administradores** | **Autenticación** | **Alta** | **Activo** | **Garantizar el acceso seguro y controlado en el sistema.** | **Crear, modificar, eliminar y ver** | **Programado** |
| **002** | **Se deberá recoger la información personal de los estudiantes, incluyendo nombre, apellidos y fecha de nacimiento.** | **Datos Personales** | **Alta** | **Activo** | **Obtener los datos necesarios del usuario y brindar mayor seguridad.** | **Registrar, Modificar y ver** | **En Proceso** |
| **003** | **Los profesores podrán ingresar las calificaciones de los estudiantes en diferentes asignaturas** | **Calificaciones** | **Alta** | **Activo** | **Brindar una mayor seguridad y confianza para el usuario** | **Ingresar, modificar y ver** | **En proceso** |
| **004** | **El Sistema deberá interactuar, con los sistemas de la UGEL, estructurando la interoperabilidad** | **Interconexión** | **Media** | **Activo** | **EL intercambio de información entre sistemas operables** | **Estandarizar la integración de la información entra sistemas distintos** | **En proceso** |
| **005** | **EL sistema deberá contar con una integridad escalables, para soportar los contantes cambios tecnológicos** | **Escalabilidad** | **Media** | **Activo** | **Garantizar el constante flujo de integración cambiante de tecnologías.** | **Poder mejorar el flujo de la modularización del sistema, para evitar colapsos.** | **En proceso** |
| **006** | **El sistema deberá contar con protección segura, encriptada y distribuida, dentro del mismo, para la validación de contraseñas, notas y datos personales o de sistema.** | **Encriptación** | **Media** | **Activo** | **Proteger la confidencialidad de la información mediante la conversión de datos en un formato ilegible para cualquier persona que no tenga la clave de descifrado autorizada** | **Es la validación de datos encriptados(ilegibles) para proteger la integridad de la información.** | **En proceso** |

1. **ENTORNO DE PRUEBAS DEDICADO:**

**La infraestructura de pruebas debe estar ubicada en un servidor compartido, separado del entorno de desarrollo y preferiblemente no accesible para el personal de desarrollo. Esto asegura la integridad de las pruebas al evitar interferencias o modificaciones accidentales por parte del equipo de desarrollo. Además, permite mantener un ambiente controlado exclusivamente para las actividades de pruebas, facilitando la detección y resolución de problemas sin afectar el desarrollo del software.**

1. **NOMBRE DE DOMINIO DISTINTO**

* **Utilizar nombres de dominio distintos a los de producción y desarrollo para evitar confusiones y garantizar que el personal de pruebas sepa claramente en qué entorno se encuentra. Esto ayuda a mantener la segregación de entornos y a evitar accidentes o modificaciones en ambientes no deseados.**

1. **SIMILITUD CON EL AMBIENTE DE PRODUCCION**

* **Asegurar que el entorno de pruebas sea lo más similar posible al ambiente de producción. Esto incluye replicar aplicaciones locales, configuraciones de servidores, manejadores de bases de datos, configuración de bases de datos, cuentas de usuario, componentes de infraestructura, versiones de software y réplicas de componentes con interoperabilidad planificada. Esta similitud garantiza una mayor fiabilidad en las pruebas y reduce los riesgos de problemas inesperados en producción.**

1. **AUTOMATIZACION DE PROPAGACION DE CAMBIOS**

* **Automatizar la propagación de cambios entre el entorno de desarrollo y el de pruebas, especialmente si no es posible tener el mismo manejador de base de datos. Utilizar herramientas de software dedicadas para esta tarea ayuda a mantener la consistencia y rapidez en la implementación de cambios, reduciendo el riesgo de errores humanos.**

1. **HERRAMIENTAS DE GESTION EN UN SERVIDOR DIFERENTE**

* **Instalar las herramientas de gestión de casos de prueba y gestión de incidencias en un servidor compartido, separado del entorno de pruebas integrales. Esto permite un acceso centralizado por parte de los equipos de pruebas y facilita la colaboración y seguimiento de actividades relacionadas con la calidad del software.**

1. **CAPACIDAD PARA MULTIPLES AUDIENCIAS**

* **Configurar el entorno de pruebas para servir a múltiples audiencias, como administradores de sistemas, usuarios finales y desarrolladores, proporcionando acceso exclusivamente para la ejecución de pruebas. Esto asegura que cada grupo de usuarios tenga las herramientas y permisos necesarios para realizar sus actividades de prueba de manera efectiva.**

1. **HERRAMIENTA DE CONTROL DE VERSIONES**

* **Apoyarse en herramientas de control de versiones para gestionar el código fuente y los cambios, especialmente cuando se realizan múltiples proyectos en paralelo para probar distintas funcionalidades. Esto permite un seguimiento preciso de las modificaciones realizadas, facilita la colaboración entre equipos y brinda la posibilidad de revertir cambios si es necesario.**

**Estas pautas son fundamentales para crear un entorno de pruebas efectivo y preciso. Al ubicar la infraestructura de pruebas en un servidor compartido y separado del entorno de desarrollo, se garantiza un ambiente controlado y aislado. Esto evita posibles interferencias o modificaciones no autorizadas que podrían afectar la integridad de las pruebas. Además, al restringir el acceso al personal de desarrollo, se mantiene la confidencialidad de los datos y se asegura que las pruebas se realicen de manera objetiva y sin influencias externas. Estas medidas son esenciales para evaluar y validar de manera precisa la funcionalidad de la herramienta de gestión de calificaciones en el contexto específico del colegio Monitor Huascar.**

1. **ANALISIS DE VIABILIDAD DE AUTOMATIZACION (¿POR QUÉ?)**

**El proceso de evaluar la viabilidad de la automatización en un proyecto, como la herramienta tecnológica para la gestión y seguimiento de calificaciones en el colegio Monitor Huascar, implica analizar detalladamente si tiene sentido y resulta beneficioso automatizar ciertos aspectos del desarrollo y las pruebas. Esto incluye considerar factores como la complejidad de las tareas a automatizar, el tiempo y recursos necesarios para implementar la automatización, los beneficios esperados en términos de eficiencia y calidad, y la capacidad de mantener y actualizar los scripts o herramientas de automatización en el futuro.**

**FASE II:**

**PLANIFICACION DE PRUEBAS**

**ACTIVIDADES**

1. **Preparación del plan de prueba / documento de estrategias para diferentes tipos de pruebas.**

* **Equipo de pruebas y roles**
* **Estrategia de las pruebas**
* **Criterios de aceptación y rechazo**
* **Criterio de suspensión y condiciones de prueba de reanudación.**

1. **Selección de herramientas de prueba**
2. **Estimación de los esfuerzos de prueba**

* **Estimación por fase**
* **Estimación por pruebas**

1. **Planeación de costos**
2. **Cronograma**

* **Proyecto prueba de software**
* **Pruebas de Software**

1. **PREPARACION DEL PLAN DE PRUEBAS**
   1. **Equipo de pruebas y roles**

|  |  |  |  |
| --- | --- | --- | --- |
| **FUNCIONES** | **DESARROLLADOR** | **ANALISTA** | **TESTER** |
| **Asegura la calidad de los entregables de pruebas** | **I** | **R** | **R** |
| **Asegura el cumplimiento de los indicadores definidos para el proceso de pruebas** | **I** | **A** | **R** |
| **Segura que los requerimientos funcionales definidos estan plenamente identificados en los escenarios de pruebas** | **I** | **R** | **R** |
| **Liderar el análisis desde el punto de vista funcional y garantizar que los entregables de desarrollo estén bien definidos y en condiciones óptimas para ser atendidos por el equipo de prueba.** | **I** | **AR** | **C** |
| **Participar activamente en el proceso de desarrollo desde el punto de vista del entendimiento funcional para identificar de forma temprana riesgos y ajustes necesarios al software.** | **I** | **AR** | **R** |
| **Diseñar los escenarios y casos de prueba que garanticen la calidad del software y el cumplimiento de los requisitos.** | **I** | **AR** | **I** |
| **Probar cada uno de los casos de prueba diseñados y ajustarlo cuando se requiera.** | **I** | **I** | **R** |
| **Documentar adecuadamente conforme con los estándares definiendo los casos fallidos del software, asi como los exitosos.** | **I** | **R** | **R** |

|  |  |  |  |
| --- | --- | --- | --- |
| **FUNCIONES** | **DESARROLLADOR** | **ANALISTA** | **TESTER** |
| **Realizar las pruebas Unitarias** | **R** | **I** | **I** |
| **Realizar las revisiones del codigo conforme con los estándares.** | **R** | **I** | **I** |
| **Realizar el ajuste de los errores reportados por el equipo de pruebas.** | **R** | **I** | **I** |
| **Capacitar y velar porque el equipo tenga un entendimiento de producto.** | **I** | **I** | **I** |
| **Identificar riesgos en el proceso de ejecución y escalar cuando sea necesario.** | **I** | **R** | **I** |
| **Gestionar la asignación de tareas dentro del equipo de pruebas.** | **I** | **I** | **I** |
| **Administrar las herramientas de gestion de pruebas.** | **I** | **I** | **I** |
| **Delegar las funciones a los demas integrantes del equipo.** | **I** | **CI** | **I** |
| **Valida constantemente el avance del proceso** | **I** | **I** | **I** |

**R = responsable**

**A = Autoridad**

**C = Consultor**

**I = Informado**

* 1. **ESTRATEGIA DE PRUEBAS**

**Se requiere que el equipo de desarrollo y el usuario certifiquen el software de Notas - Sistema de Gestión de Notas en tres etapas, asegurando que administre y gestione eficazmente los procesos de la escuela Monitor Huascar.**

* **En la primera etapa del proceso de certificación del software de gestión de notas, se verificará la funcionalidad de los módulos de Registrar Alumno, Actualizar Alumno y Eliminar Alumno. Esto implica evaluar que el sistema permita ingresar nuevos alumnos de manera correcta, actualice la información de los alumnos de forma precisa y segura, y elimine registros de alumnos adecuadamente según las políticas establecidas.**
* **En la segunda etapa del proceso de certificación del software de gestión de notas, se evaluarán las funcionalidades integradas de los módulos de Alumno para añadir nota, promediar y consultar si el alumno pasa de año o no.**
* **En la tercera etapa del proceso de certificación del software de gestión de notas, se evaluará la funcionalidad de las cuentas de usuario para los profesores. Esto implica verificar que el sistema permita a los profesores acceder de manera segura al sistema de gestión de notas y que tengan las capacidades necesarias para realizar sus tareas específicas. Estas tareas pueden incluir la introducción de calificaciones, la generación de informes de progreso académico, el seguimiento del rendimiento de los alumnos, entre otros.**

**El objetivo es realizar pruebas exhaustivas en los módulos de gestión y administración de notas para asegurar su funcionalidad y eficacia. Estas pruebas incluyen:**

* **Ingreso al sistema y creación de datos (Alumnos, Aulas y Profesores).**
* **Creación, modificación y eliminación de Alumnos, Aulas y Profesores.**
* **Verificación de la promediación de las notas de los Alumnos.**
* **Creación de reportes de Notas del Alumno.**
* **Creación de consultas de documentos por rango de fechas por parte de los profesores.**
* **Revisión y aprobación del director en caso de pasar o no de año.**

**El objetivo final es verificar y validar los resultados obtenidos en cada uno de estos subobjetivos, asegurando que el software cumpla con los requisitos y expectativas del colegio Monitor Huascar en la gestión y administración de las notas de los estudiantes.**

* 1. **CRITERIOS DE ACEPTACION Y RECHAZO**

|  |  |  |
| --- | --- | --- |
| **NOMBRE INDICADOR** | **EDICION** | **UMBRAL DE TOLERANCIA** |
| **Fallos Críticos** | **Cantidad de errores donde los usuarios no pueden utilizar las funcionalidades principales del sistema.** | **Se aprobará el proyecto con un 100% de las pruebas ejecutadas, pero con un 90% de aceptación. Esto quiere decir el 90% de las pruebas deben ser exitosas y sin errores, fallas, defectos. El restante 10% pueden existir errores medios o bajos, pero no graves. En caso de ocurrir que el proyecto no cumpla con el nivel exigido, el proyecto se rechaza completo en su etapa de certificación.** |
| **Fallos Mayores** | **Cantidad de errores donde el sistema opera con restricciones que impiden completar la operación de negocio que define el requerimiento.** |
| **Fallos Medios** | **Cantidad de errores donde no se encuentran disponibles algunas funciones o componentes del sistema, que generan un impacto mínimo para los usuarios del sistema.** |
| **Fallos Cosméticos** | **Cantidad de errores en la interfaz del usuario que no impide la correcta ejecución del sistema.** |

* 1. **CRITERIOS DE SUSPENSION Y REANUDACION DE PRUEBAS.**

**Los criterios de suspensión de las pruebas se establecen en función de la identificación y evaluación de riesgos que no han sido mitigados adecuadamente, que no fueron identificados previamente o que se materializaron durante el proceso de pruebas. Estos criterios son fundamentales para determinar si las pruebas deben detenerse temporalmente o incluso cancelarse por completo si se considera que la calidad del software o la integridad del proceso de pruebas están comprometidas.**

**A continuación, se relacionan a manera de ejemplo, algunas situaciones que pueden obligar la suspensión de las pruebas de forma parcial o total y que pueden hacer parte de la matriz de riesgos del proyecto en la cual se deben incluir los correspondientes a las pruebas, para poder establecer los planes de mitigación y tenerlos controlados:**

* **No se cuenta con el Software disponible en las fechas establecidas para el inicio de la etapa de pruebas. La condición para reanudar obedecerá a que el equipo de pruebas cuente con las funcionalidades dispuestas en el ambiente de pruebas establecido.**
* **Las funcionalidades se encuentran incompletas impidiendo poder ejecutar un flujo básico de información. La condición para reanudación obedecerá a poder contar con las funcionalidades y la evidencia de las pruebas unitarias por parte del equipo de desarrollo.**
* **Existe un fallo crítico que impide ejecutar la funcionalidad y bloquea el proceso de pruebas. La condición para reanudación obedecerá a la resolución de fallo.**
* **Existen cambios en la definición del alcance en las pruebas. La condición para reanudar obedecerá al cierre del alcance de las pruebas.**

1. **SELECCIÓN DE HERRAMIENTAS PARA LAS PRUEBAS**

**La herramienta principal que utilizamos para asegurar la calidad y el rendimiento en las pruebas del sistema de gestión de notas del colegio es NUnit. Se trata de un marco de prueba unitaria sólido y adaptable diseñado específicamente para el entorno .NET. NUnit simplifica la creación y ejecución de pruebas unitarias, garantizando que cada componente de nuestro proyecto funcione correctamente y cumpla con los requisitos establecidos.**

**En conjunto con NUnit, hemos incorporado Moq, una biblioteca de simulacros potente que nos permite llevar a cabo pruebas unitarias de manera eficiente al simular el comportamiento de objetos y componentes. Esto asegura la coherencia y la fiabilidad de nuestro sistema durante las pruebas.**

**Para abordar las pruebas de comportamiento, hemos integrado SpecFlow, una herramienta de prueba de comportamiento (BDD) que nos permite describir y ejecutar escenarios de prueba en un formato natural y fácil de entender. Esta integración facilita la colaboración entre equipos y asegura que las características críticas de nuestro sistema sean validadas de manera efectiva.**

**Además, utilizamos Visual Studio Test Explorer como una interfaz gráfica de usuario para ejecutar y analizar nuestras pruebas de manera eficiente dentro del entorno de desarrollo de Visual Studio. Esto proporciona una integración fluida con nuestro proceso de desarrollo y nos permite realizar un seguimiento detallado del rendimiento de nuestras pruebas.**

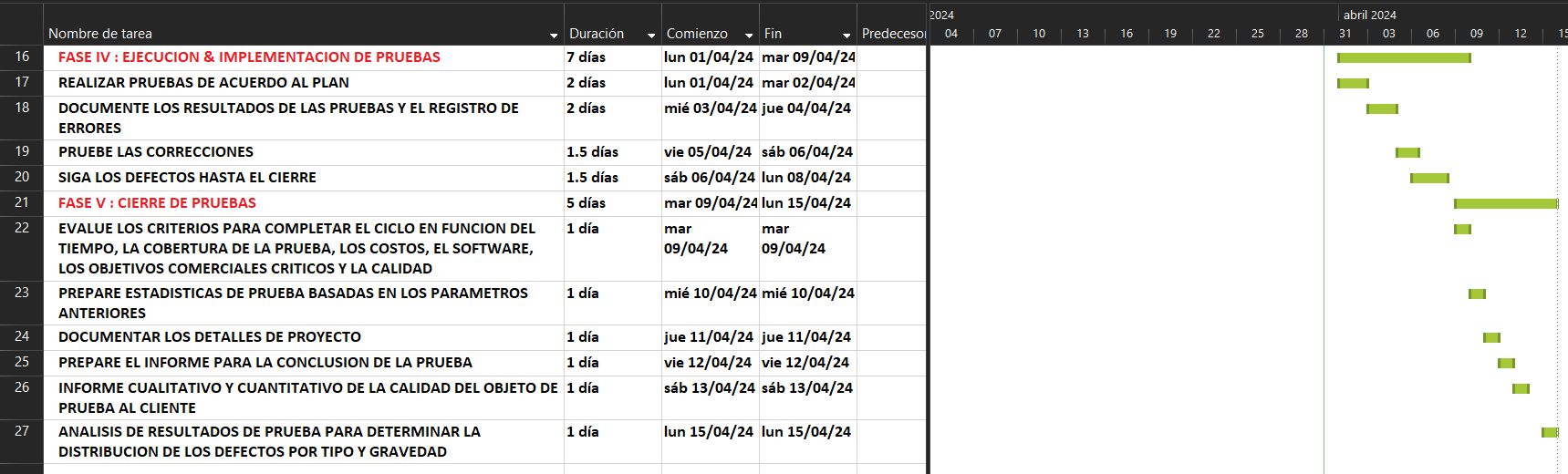
1. **ESTIMACION DE ESFUERZO DE LAS PRUEBAS (MODELO PERT)**
2. **PLANEACION DE COSTOS**
   1. **HARDWARE Y DISPOSITIVOS**
   2. **SOFTWARE**



* 1. **PERSONAL**
  2. **COSTO MATERIALES**



* 1. **COSTOS TOTALES**

1. **CRONOGRAMA (MICROSOFT PROYECT)**
   1. **CRONOGRAMA PROYECTO PRUEBA SOFTWARE**
   2. **CRONOGRAMA PRUEBA DE SOFTWARE**

****

**FASE III:**

**DESARROLLO DE CASOS DE PRUEBA**

**ACTIVIDADES**

1. **Hacer casos de prueba, scripts de automatización**
2. **Revisar y preparar tambien casos de pruebas y scripts de referencia**
3. **Crear datos de prueba**

**1.PRUEBAS FUNCIONALES**

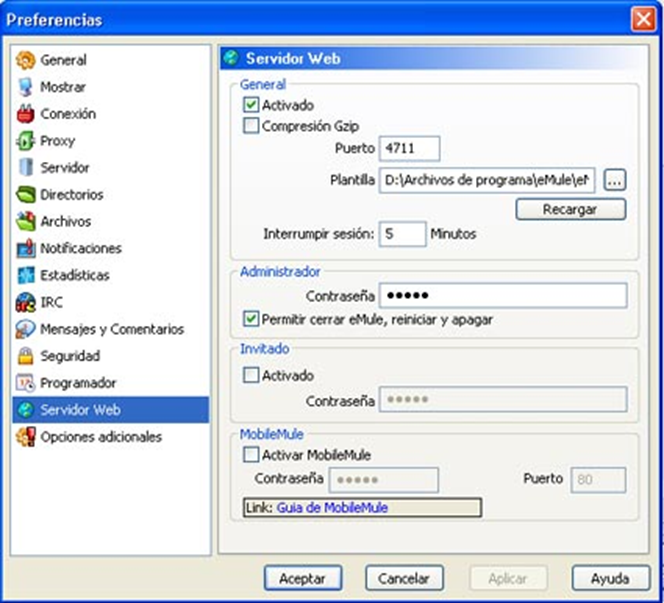
* **CASO DE PRUEBAS UNITARIAS**

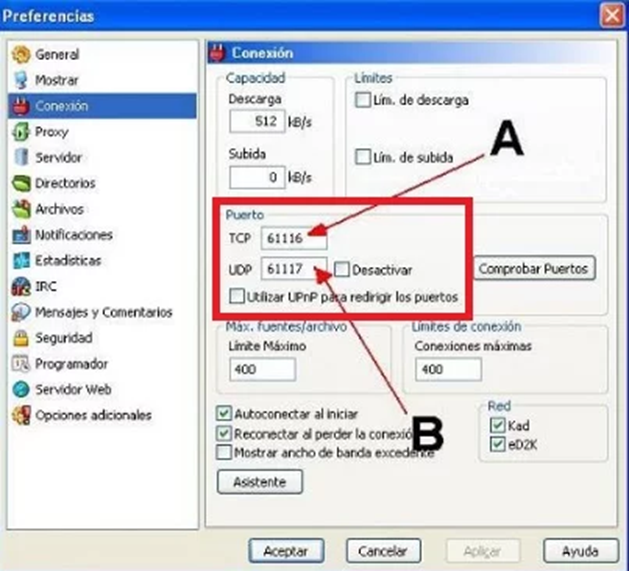
**FASE IV:**

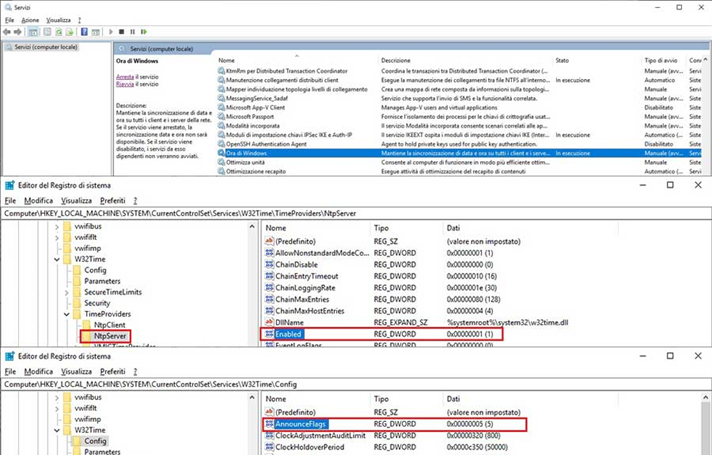
**CONFIGURAR EL ENTORNO DE PRUEBAS**

**ACTIVIDADES**

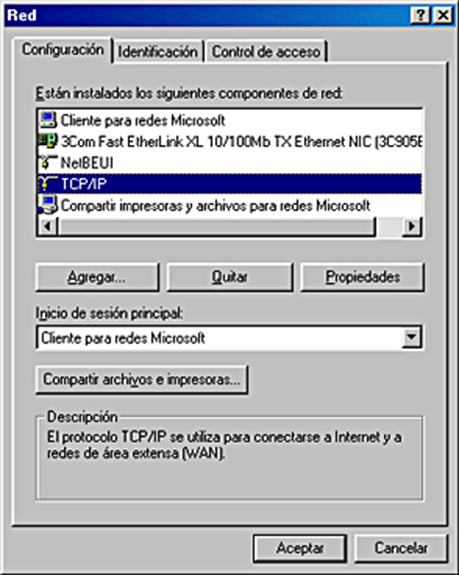
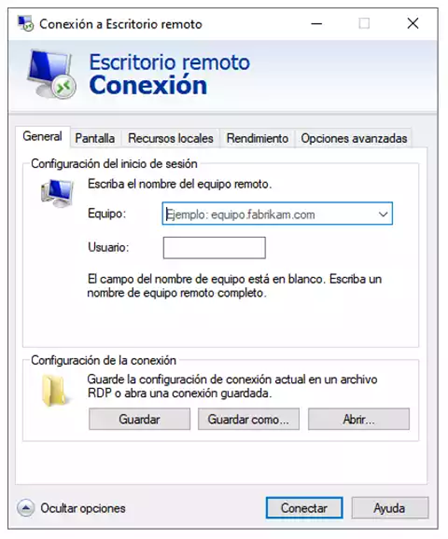
1. **Comprenda la arquitectura y el entorno requeridos y también enumere los requisitos de hardware y software para el entorno de prueba.**
2. **Prueba de configuración del entorno y los datos de prueba.**
3. **Realice una prueba de humo en la construcción.**
4. **COMPRENDA LA ARQUITECTURA Y EL ENTORNO REQUERIDO Y TAMBIEN ENUMERAR LOS REQUISITOS DE HARDWARE Y SOFTWARE PARA EL ENTORNO DE PRUEBA.**
   1. **CONFIGURACION DE SERVIDOR WEB**





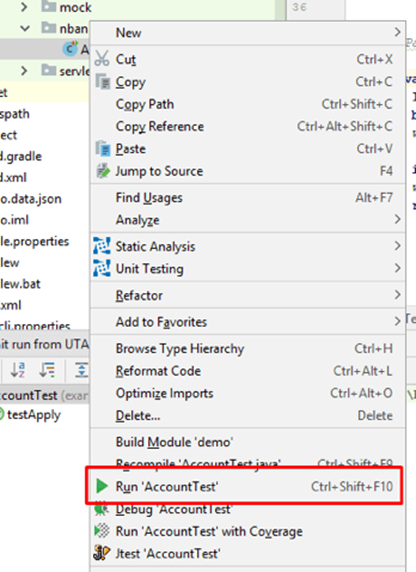
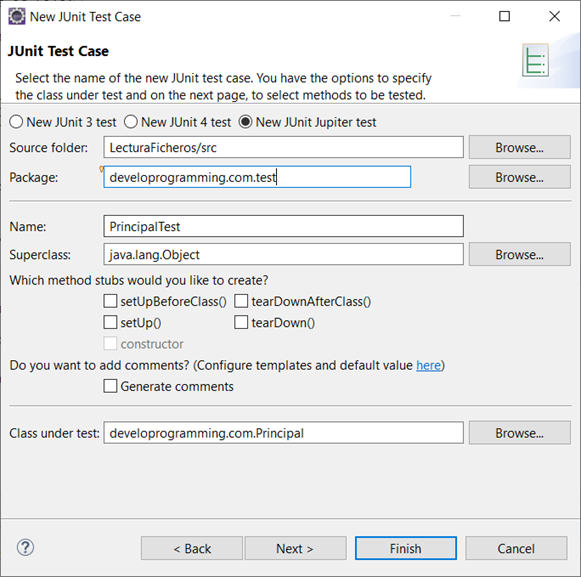


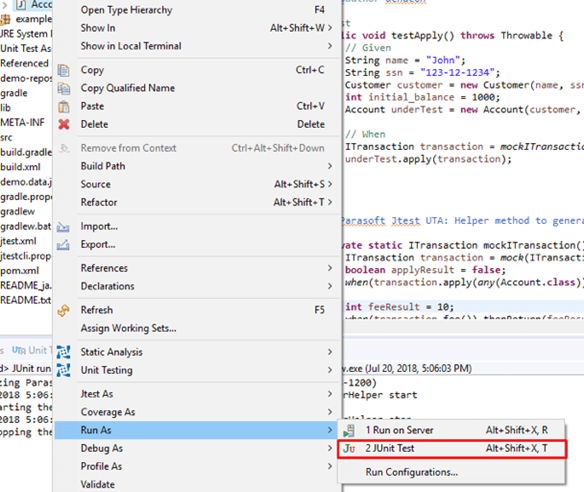
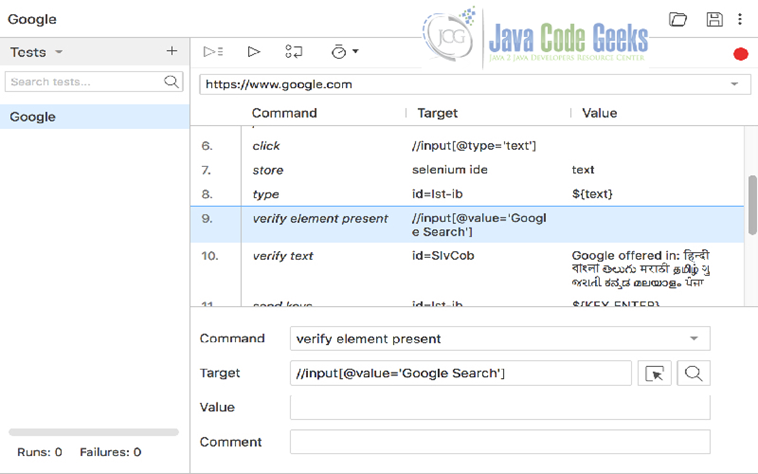
* 1. **CONFIGURACION DE PCS**

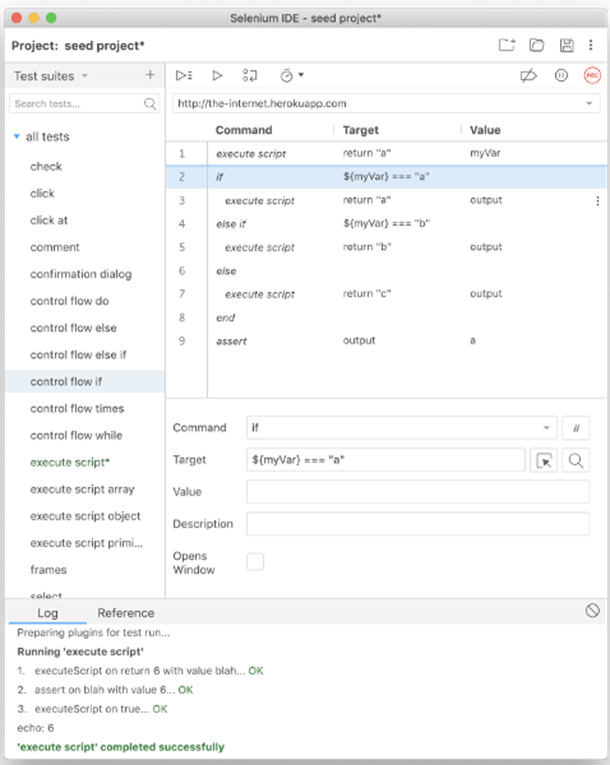




* 1. **CONFIGURACION JNUIT**



* 1. **CONFIGURACION SELENIUM**



* 1. **Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

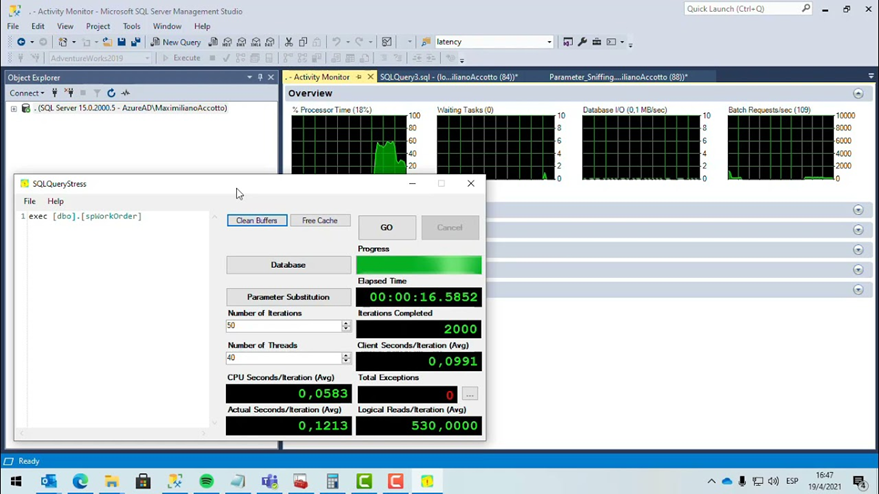
     Descripción generada automáticamenteCONFIGURACION JMETER**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

* 1. **CONFIGURACION SQLQUERY STRESS**

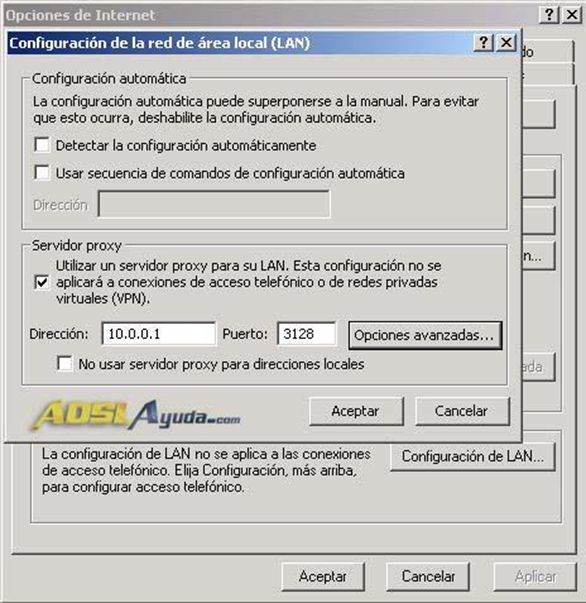
**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

* 1. **CONFIGURACION RED LAN**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



* 1. **AMBIENTE DE PRUEBAS DE SOFTWARE**



* 1. **DESCRIPCION DEL HARDWARE**



* 1. **DESCRIPCION DE LA RED**

**FASE V:**

**EJECUCION DE LOS CASOS DE PRUEBAS**

**ACTIVIDADES**

1. **Realizar los casos de pruebas de acuerdo con el plan.**
2. **Registro de informe de errores**
3. **Pruebe las correcciones**
4. **REALIZAR LOS CASOS DE PRUEBA DE ACUERDO CON EL PLAN**



* 1. **PRUEBAS UNITARIAS**



* 1. **REGISTRO DE INFORMES DE ERRORES**