1. Título
2. Agenda
3. Objetivos de las pruebas performance es la identificación temprana de los riesgos que pueden impactar de forma negativa la experiencia del usuario final.
   1. Tiempo de respuesta (Response time): Criterio de aceptación
   2. Cantidad de trabajo por unidad de tiempo (Throughput): Ejemplo TPS
   3. Utilización de recursos (Resource Utilization): Servidores que hagan parte de la infraestructura.
   4. Carga de trabajo (Workload): Distribución de carga en cada una de las transacciones.
   5. (No se busca determinar, encontrar o indagar en la naturaleza de problemas)
4. Beneficios:
   1. Técnicos y Operativos: Identificar el comportamiento en los tiempos de respuesta. Conocer la experiencia del usuario al interactuar en distintos escenarios. Identificar cuellos de botella en la arquitectura o componentes. Facilita el afinamiento de los componentes. Entrega información que facilita la toma de decisiones al negocio.
5. Casos recomendables para las pruebas de performance:
   1. Tiempo de respuesta que afecta la usabilidad de la aplicación.
   2. Si la aplicación será utilizada por múltiples usuarios simultáneamente.
   3. La aplicación comparte infraestructura con otras aplicaciones.
   4. Si la aplicación tiene una arquitectura distribuida. (OnPremise / Nube)
   5. Se realiza una nueva modificación o se implementa una nueva versión.
   6. Se cuenta con un gobierno que abarca pruebas funcionales y no funcionales.
6. Que se intenta responder con las pruebas:
   1. ¿El sistema soporta la carga esperada o sostenida?
   2. ¿El sistema se recupera después de la sobrecarga?
   3. ¿Cuáles de los componentes presentan fallas o reducción de rendimiento en la experiencia de usuario?
   4. ¿El sistema es escalable?
   5. ¿Las modificaciones, nuevos aplicativos o usuarios adicionales tendrán un impacto en el rendimiento?
7. Pruebas performance y NO Funcionales
   1. De acuerdo a la ISO/IEC 25010 se tiene 7 atributos no funcionales, las pruebas performance abarcan 2
      1. Fiabilidad: Madurez, Tolerancia a fallos, Capacidad de recuperación, Disponibilidad o Estabilidad.
      2. Eficiencia: Capacidad, Comportamiento en el tiempo, Utilización de recursos.
8. Conceptos generales parte 1:
   1. Transacción: Grupo de operaciones que realiza un usuario en un aplicativo para ejecutar una función. Ejemplo: Consultar saldo (Back).
   2. Interacción: Acción funcional que realiza un usuario con el sistema. Desde el punto de vista del usuario es una transición de pantalla a otra (Front).
9. Conceptos generales parte 2:
   1. Criterios de aceptación: Valores razonables y medibles de los objetos de performance que son admitidos como el rendimiento aceptable para el aplicativo.
   2. Criterio: Velocidad en las transacciones desde el punto de vista del usuario o de un componente. Número esperado de usuarios o transacciones por unidad de tiempo. Duración máxima del pico. Nivel de tolerancia en tiempo de respuesta.
10. Conceptos generales parte 3:
    1. Tiempos de respuesta (Response Time): El tiempo que pasa desde que se envía una comunicación y se recibe la respuesta. Cuanto tarda la aplicación en realizar las operaciones que el usuario necesita.
    2. Tiempos de espera (Think Times): Es lo que el usuario tarda en realizar las acciones que el sistema está esperando para efectuar sus operaciones. Tiempo de pensado.
11. Conceptos generales parte 4:
    1. Cantidad de trabajo por unidad de tiempo (Throughput): Es la cantidad de transacciones u operaciones que se realizan por unidad de tiempo, generalmente por 1 segundo.
    2. Concurrencia: Cantidad de usuarios que se conectan a una aplicación y realizan operaciones al mismo tiempo. Transacciones no es igual a los usuarios, un usuario puede hacer más de una transacción.
    3. Simultaneidad (No se puede simular): Requiere que todos los usuarios estén haciendo lo mismo en el mismo punto.
    4. Script: Conjunto de instrucciones para ser ejecutadas paso a paso y que nos permite automatizar tareas.
12. Conceptos generales parte 5:
    1. Fallo: Desviación del componente o resultado esperado. Cuando se materializa un fallo en un ambiente.
    2. Defectos: Una imperfección de un componente o sistema que puede causar que el sistema falle al momento de desempeñar sus funciones. Lo que está mal (Pruebas estáticas).
    3. Error: Lo comete las personas o las condiciones medioambientales.
       1. Humanos: Errores de programación, obviando los requisitos del producto, presión del tiempo, Falta de experiencia o comunicación, malos entendidos, ambigüedades en los requerimientos.
       2. Condiciones medioambientales: Radiación, contaminación, cambio en el fluido eléctrico.
    4. Ventana de ejecución: Periodo en el que se acuerda con el equipo llevar a cabo la ejecución de un proceso o prueba.
    5. Distribución de Poisson: Se usa para identificar la ocurrencia de tener un numero de usuarios concurrentes en el sistema.
13. Conceptos generales parte 6:
    1. Utilización de recursos (Resources Utilization): Los recursos son aquellos componentes de hardware que apoyan el rendimiento de la aplicación. (Procesador, Memoria, Disco Duro, Ancho de Banda.
    2. Cuellos de botella (Bottlenecks): Es un punto de la aplicación donde ocurre congestión y demora, retrasando el procesamiento de las peticiones.
14. Conceptos generales parte 7:
    1. Puntos lentos (Slow Points): Son elementos de la aplicación que tardan un gran tiempo en entregarse al usuario como gráficos muy pesados, a diferencia de los cuellos de botella no tienen efectos sobre otras áreas de la aplicación, ni tienen que ver con componentes que operen por debajo de la capacidad necesaria.
    2. Herramientas de Performance: Jmeter, Wapt Pro, Neoload.
15. Conceptos generales parte 8:
    1. El usuario percibe el rendimiento de la aplicación al hacer uso de la misma (front); sin embargo, las tareas las ejecuta el Back. El front realiza una serie de peticiones y las recibe el servidor.
       1. Front-end: Elementos de página, visualización de componente.
       2. Back: Funciones y métodos que se ejecutan,
       3. Middleware: Son las conexiones de esas dos por buses.
    2. Tipos de Cloud Computing:
       1. SaaS: Usuario Finales.
       2. PaaS: Desarrolladores de aplicaciones.
       3. IaaS: Ingeniero de redes, Administradores técnicos.
    3. Tipos de arquitectura:
       1. Cloud
       2. On Premise.
       3. Híbridas.
    4. Realizar estrategias de pruebas.
16. Conceptos generales parte 9:
    1. Tanto el Middleware como el Backend puede ser consumidos directamente por herramientas de performance o especializadas como Postman o SoapUI.
       1. Postman me deja enviar varios usuarios secuencialmente, no con concurrencia.
       2. SoapUI: Se enfoca a los servicios SOAP, aunque se puede usar con Rest.
    2. Sintaxis: Protocolo + Servidor + Puerto + Path
       1. Si no hay “Dos Puntos” se trabajan con puertos por defecto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROTOCOLO | DESCRIPCIÓN | SEGURIDAD | PUERTO | VELOCIDAD |
| HTTP | HyperText Transfer Protocol | No es seguro | 80 | Ligeramente más veloz |
| HTTPS | HyperText Transfer Protocol Secure | SSL/TLS (Secure Sockets Layer/Transport Layer Security) | 443 | El cifrado tiene un mayor retardo |

* 1. Métodos de en servicios:

|  |  |
| --- | --- |
| SOAP | REST |
| getCustomerDetails: Recupera los detalles de un cliente específico. | GET: Recupera recursos del servidor |
| updateOrderStatus: Actualiza el estado de una orden. | POST: Crea un nuevo recurso |
| createInvoice: Genera una nueva factura para una orden. | PUT: Actualiza un recursro existente o crea uno en caso de que no exista, |
| deleteProduct: Elimina un producto del inventario | DELETED: Elimina un recurso del servidor. |
|  | PATCH: Realiza una actualización parcial del recurso. |

1. Conceptos generales parte 10
   1. Proxy: Intermediario de realización de petición entre un cliente y un servidor.
   2. Con el script permite tener datos estadísticos, simulación de usuarios, datos de rendimiento, datos de los recursos utilizados y realizar un mejor testplan.
2. Conceptos generales parte 11
   1. Tener en cuenta las tecnologías.
      1. App, SOAP, Móvil, Web
   2. Flujo de transacción:
      1. Procesos de negocios y alcance.
      2. Diagrama de arquitectura.
   3. Ambiente:
      1. Desarrollo:
      2. Calidad: Pocos recursos.
      3. Pre producción Stage: Más recursos.
      4. Producción: Generan riesgos muy altos como afectación de clientes reales (Habeas data). Situaciones muy puntuales, fallos que solo se presenten en Producción.
      5. No hay que extrapolar datos. Suponer resultados con base de un ambiente a otro.
3. Tipos de prueba performance parte 1
   1. Una solución integral consta de consumir Web Services, API y Componentes haciendo transacciones de negocio integrales y E2E.
4. Tipos de prueba performance parte 2
   1. Pruebas de rendimiento (Pruebas Performance)
      1. Línea base (Base line): Se enfoca en tener una primera aproximación al rendimiento general del sistema.
      2. Pruebas de carga (Load test): Verificar el comportamiento del sistema cuando se somete a una cantidad especifica de usuarios en un momento determinado.
      3. Pruebas de estrés (Stress test): Buscar algún tipo de saturación en algunos de los componentes, y ejecutar una carga superior a la que el sistema está acostumbrado.
5. Tipos de prueba performance parte 3
   * 1. Pruebas de escalabilidad (Scalability test): Identificar las capacidades de recuperación en el uso de recursos de plataforma y la capacidad de rendimiento del sistema bajo la demanda de diferentes cantidades de concurrencia de usuarios, tanto en alta como en baja demanda, con el fin de hacer su proyección a futuro.
     2. Pruebas de pico (Peak test): Verificar el funcionamiento del sistema cuando se somete a cargas extremas varias veces durante un corto periodo de tiempo.
   1. Pruebas de disponibilidad:
      1. Pruebas de larga duración o estabilidad (Stability test): Evaluar cómo se comporta el sistema bajo una carga constante de usuarios durante un período prolongado. Seis u Ocho horas de ejecución.
      2. Pruebas de resistencia (endurance): El fin de estas pruebas es comprobar si el sistema tiene la capacidad de soportar una carga de forma continua. Seis u Ocho horas de ejecución.
      3. Pruebas de tolerancia a fallos o resiliencia (resilience test): Evaluar la capacidad del sistema para recuperarse de fallos inesperados en sus componentes. Crear un fallo apropósito, desconexión de un componente y luego volverlo a encender para que si se recupera.
6. Tipos de prueba performance parte 3
   1. Otras pruebas:
      1. Pruebas de volumetría (batch): Son transacciones que no tienen una concurrencia alta, pero que sí llevan información muy grande
      2. Complemento de experiencia de usuario: Transacciones que sean por front, si no se tiene alcance de front, no tiene sentido. Pruebas UX: APDEX (Application Performance Index o índice de rendimiento de aplicaciones): Aspectos de performance, accesibilidad y disponibilidad de la aplicación. Buenas prácticas a nivel de código. Tiempos de espera.
7. Tipos de prueba performance parte 4
   1. Performnace Tunning: Durante la ejecución deben estar las personas de monitoreo, redes, desarrollo; desde su capa observa lo que va asando. Es bueno para el afinamiento de componentes.
   2. Performance Express: Transacción puntual, entrega de informe preliminar.
8. Tipos de prueba performance parte 5
   1. Tipos de carga
      1. Carga escalonada
      2. Carga constante
      3. Carga en picos
      4. Se pueden combinar las cargas
9. Modelo de Servicio de Pruebas Performance Parte 1
   1. Traditional Performance Testing
      1. Servicio fijo: Tiene una persona fija por cierto tiempo que va a realizar cierto tipo de alcance, mantener al cliente informado.
      2. Servicio por proyecto: Se realiza el proyecto puntual, se agregan más horas necesarias en caso de ser necesarios.
      3. Performance Express: Fechas cortas del Date To Market, un informe no tan profundo.
      4. Performance Tunning: Se trabaja bajo una mesa técnica, ejecutar en tiempo real desde diferentes frentes
      5. Complemento de experiencia: ¿Cómo lo ve el usuario final? Sitespeed
   2. Continuous Performance Testing
      1. Servicio fijo: Tiene una persona fija por cierto tiempo que va a realizar cierto tipo de alcance, mantener al cliente informado.
      2. Performance Tunning: Se trabaja bajo una mesa técnica, ejecutar en tiempo real desde diferentes frentes
      3. Complemento de experiencia: ¿Cómo lo ve el usuario final? Sitespeed
10. Modelo de Servicio de Pruebas Performance Parte 2
    1. Fase comercial (Entendimiento):
       1. Pre-venta: Corresponde al acercamiento comercial de los clientes y finaliza con la entrega de la propuesta y se coordina la fecha de inicio del servicio, las actividades son el alcance, cobertura, servicio, la seguridad.
          1. Contextualización y entendimiento de la necesidad.
          2. Definir el alcance de las pruebas.
          3. Requisitos definidos.
             1. Propuesta de servicios.
             2. Seguimiento de servicios.
       2. Planeación:
          1. Datos de prueba.
          2. Infraestructura de pruebas.
          3. Criterios de aceptación.
          4. Alcance y estrategia.
             1. Plan de pruebas de Performance.
    2. Fase de operación (Pruebas performance):
       1. Diseño de las pruebas:
          1. Especificación de los escenarios de pruebas.
          2. Escenarios de prueba.
          3. Guiones de prueba.
          4. Automatización de pruebas.
             1. Script de pruebas.
       2. Ejecución de pruebas.
          1. Configuración de ambiente.
          2. Línea base: Consiste en ejecutar cada petición con un único usuario.
          3. La ejecución se hace de forma gradual, usualmente de 20%, 40%, 60%, 80%.
          4. El monitoreo se hace sobre los indicadores
             1. Reporte de ejecuciones.
    3. Fase de cierre y resultados (Análisis de resultados):
       1. Análisis de resultados.
          1. Re-ejecución de pruebas, cuando se implementaron cambios al haberse encontrado oportunidades de mejora.
             1. Informe de cierre o resultados: Se realiza una vez se haya alcanzado los objetivos, se detallan los resultados.
11. Modelo de Servicio de Pruebas Performance Parte 3
    1. Fase de entendimiento: Tiene como objetico identificar la necesidad del cliente frente a las pruebas de performance y plantear una estrategia que permita dar solución y cumplir las expectativas (Aprobación de la propuesta)
       1. Sesión de contexto.
       2. Coordinación de la adherencia técnica.
       3. Creación de la estrategia acorde a la necesidad.
       4. Construir la estimación y la propuesta del servicio
       5. Enviar la aprobación por parte del cliente.
    2. Fase de Pruebas performance: La operación de la ejecución de las pruebas puede estar enmarcada bajo un esquema tradicional por proyecto o bajo un servicio fijo, dependiendo de las características y estrategias del servicio planteado.
       1. Realizar el ciclo de desarrollo de pruebas
       2. Ajustar la estrategia y el plan de pruebas
       3. Coordinar y gestionar los requisitos y condiciones
       4. Realizar la ejecución
    3. Fase de análisis y resultados: Considerando el comportamiento de la infraestructura y arquitectura, generando entendimiento y recomendaciones que permitan afinar la solución tecnológica de nuestro cliente.
       1. Recolección de resultados.
       2. Correlacionar los resultados esperados con los obtenidos.
       3. Construir un informe de resultados.
       4. Presentar y comparten los resultados con los equipos.
12. Modelo de Servicio de Pruebas Performance Parte 4
    1. Pre-venta:
       1. Descripción:
          1. Contextualización inicial.
          2. Primer punto de contexto.
          3. ¿Por qué tengo que hacer esto?
          4. ¿Para qué?
          5. ¿Cuándo se necesita?
       2. Documentos:
          1. Documento de adherencia: Hacerla con todas (JMeter, WAPt Pro, Neoload, SiteSpeed.io) y con la que dé mejor resultados, hacerla con ese y recordar el licenciamiento. Preguntar si ya se hizo antes para no hacer reprocesos. (Avisar a Crisitian Vandebenito, Ronni, Wilson Pino)
          2. Cálculo de concurrencia
          3. Calculadora de estimación
          4. Propuesta comercial
    2. Planeación:
       1. Descripción:
          1. Sesión de revisión detallada del alcance
          2. Estrategia detallada
          3. Plan de pruebas
       2. Documentos:
          1. Análisis de riesgos
          2. Cronograma
          3. KickOff
    3. Diseño:
       1. Descripción:
          1. Script de performance.
          2. Habilitar el testware necesario.
          3. Solicitar y gestionar los requisitos de prueba
       2. Documentos:
          1. Script de pruebas en el repo
          2. Dimensionamiento
    4. Ejecución:
       1. Descripción:
          1. Preparación y coordinación de requisitos.
       2. Documentos:
          1. Correo con resultados parciales.
          2. Minutograma de ejecución
    5. Análisis y cierre:
       1. Descripción:
          1. Recolección de monitoreo
          2. Correlación de resultados.
          3. Cierto tiempo de guardado de la información.
          4. Criterios de aceptación
       2. Documentos:
          1. Informe de resultados (General y Técnico)
    6. Re test:
       1. Descripción:
          1. Revisión de preparación del testware.
          2. Informe de resultados actualizado.
          3. Se entiende que debe estar al 80% de pruebas funcionales.
          4. Se puede solicitar un informe preliminar, en caso de que salga errores para hacer un retest.
          5. Contar los hallazgos en el informe final.
       2. Documentos:
          1. Informe actualizado de resultados
13. Modelo de Servicio de Pruebas Performance Parte 4
    1. La adherencia es el proceso inicial de Performance el cuál evalúa la complejidad de la app y en dónde se desea probar el rendimiento.
    2. Grabar y reproducir la transacción o alcance deseado.
    3. Recolectar métricas e indicadores de rendimiento.
    4. Generar visibilidad de la automatización
    5. Simular operaciones de los usuario o flujos.
    6. Herramientas:
       1. JMeter
       2. Wapt Pro
       3. Neoload
       4. SiteSpeedIO
14. Modelo de Servicio de Pruebas Performance Parte 5
    1. Desarrollo:
       1. Planteamiento -> Análisis y diseño -> Desarrollo & PU
       2. Estimación de esfuerzo
       3. Aprobación
    2. Pruebas / Preproducción
       1. Pruebas integrales / Pruebas UAT
       2. Planeación (Análisis, Diseño y Ejecución)
       3. Ajuste y Ejecución
       4. Alertas o incidentes
    3. Producción
       1. Mejora continua
       2. Iniciar antes del 80% con diseños y construcción tiene el riesgo de reprocesos
15. Modelo de Servicio de Pruebas Performance Parte 6
    1. CPT: Adherencia de los procesos de los clientes
    2. Hacer un valor de pruebas tempranas
16. Modelo de Servicio de Pruebas Performance Parte 7
    1. La arquitectura puede estar en aplicaciones o en la nube
    2. Realizar estrategia de pruebas
    3. Auditar y desplegar las pruebas de performance
    4. Primero la línea base
    5. Ejecutar rutas críticas
    6. Validar cumplimiento de los criterios de aceptación
    7. Socializar los resultados
17. Mejores Prácticas parte 1
    1. Cada prueba es diferente a otra
    2. Objetivos de las pruebas: Motivaciones o expectativas
    3. Criterios de aceptación: Qué espera el cliente
    4. Ambiente de la prueba: Pueden tener varios ambientes de pruebas
    5. Data: Toda la data que se requiere. En especifica o Data reutilizable y el volumen.
    6. Monitoreo: Contar con un monitoreo de la plataforma tecnológica que permita correlacionar la información
18. Mejores Prácticas parte 2
    1. Objetivos de negocio:
       1. Características operativas
       2. Capacidad de usuarios
       3. Tiempos de respuesta
       4. Criterios de aceptación
    2. Estrategias a plantear:
       1. Escenarios realistas
       2. Simulación de condiciones
       3. Estrategias orientadas al negocio
    3. Análisis de resultados:
       1. Resultados eficientes
       2. Enfoque del riesgo
       3. Evidencias cuantificadas