PERFORMANCE

1. ¿Qué son las pruebas de performance?

* Es una serie de técnicas, herramientas y estrategias que evalúan la calidad del software en cuestión de eficiencia (utilización de recursos) y fiabilidad (Recuperación y tolerancia a fallos). ISO/IEC 25010.
* Evaluar la capacidad de procesamiento, el tiempo de respuesta, el comportamiento y la utilización de recursos tecnológicos.
* Como responde el sistema, qué tan bien y qué tan rápido.
* Las pruebas de performance son las que determinan lo rápido que realiza una tarea un sistema en condiciones particulares.
* Según las estrategias enfocadas en el negocio, aplicar las pruebas de performance.
* Estrega de información estratégica.
* Toolbox: Contar con herramientas de software libre o licenciadas para dar mayor cobertura.

1. Objetivos:

* Minimizar riesgos de desempeño, rendimiento y disponibilidad sobre el activo digital o algún componente.
* Identificar puntos de congestión en la arquitectura.
* Socializar los resultados y entregar las recomendaciones desde el punto de vista de performance.
* Identificar y proponer escenarios con condiciones de uso en los que el software pueda verse enfrentados en la vida real, simulados con data de prueba.
* Tiempo de respuesta (Response time) / Cantidad de trabajo por unidad de tiempo (Throughput) / Utilización de recursos (Resources Utilization) / Carga de trabajo (Workload)

1. Beneficios:

* Entregar información para mejorar la toma de decisiones de negocio y a TI.
* Identificar el comportamiento y los tiempos de respuesta del aplicativo.
* Evidencia la experiencia del usuario frente a los escenarios de performance.
* Permite identificar los cuellos de botella en a la arquitectura.
* Facilitan el afinamiento de la estructura.

1. Activación de pruebas:

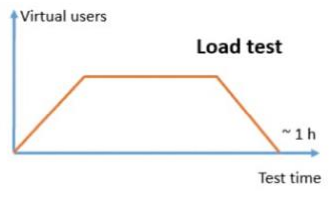
* Cuando el tiempo de respuesta afecta la usabilidad.
* Cuando la aplicación es utilizada por varios usuarios al mismo tiempo.
* Cuando la aplicación comparte arquitectura con otras aplicaciones.
* Cuando la aplicación tiene una arquitectura distribuida.
* Cuando se realiza una nueva funcionalidad o nueva versión.
* Cuando se tiene un gobierno de calidad con pruebas funcionales y no funcionales.

1. Alcance:

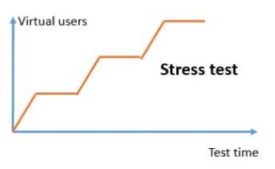
* Las pruebas de performance no encentrar la causa raíz del problema.
* APDEX: es un indicador utilizado a nivel mundial para evaluar y medir la satisfacción de los usuarios en relación con el rendimiento de los canales digitales.
* ¿El sistema soporta la carga esperada?
* ¿El sistema soporta carga sostenida?
* ¿El sistema se recupera después de una sobrecarga?
* ¿Cuál es de los componentes presentan fallas y/o la reducción de rendimiento en la experiencia del usuario final?
* ¿Es el sistema escalable, en número de usuarios o transacciones?
* ¿Las modificaciones, nuevos aplicativos o usuarios adicionales tendrán un impacto en el rendimiento?​​​​​​​

1. Tipos de pruebas

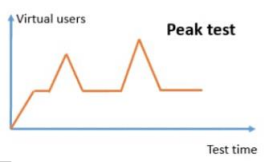
* Pruebas de rendimiento (Pruebas performance):
  + Objetivo:
    - Saber cuántos usuarios se soporta.
    - Brindar información para mejorar tiempos de repuesta.
  + Pruebas de carga (Load test): Verificar el comportamiento del sistema cuando se somete a una cantidad especifica de usuarios en un momento determinado.



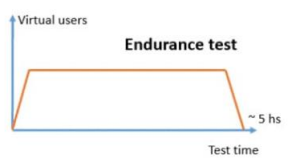
* + Pruebas de estrés (Stress test): Buscar algún tipo de saturación en algunos de los componentes, y ejecutar una carga superior a la que el sistema está acostumbrado.



* + 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.
  + Pruebas de pico (Peak test): Verificar el funcionamiento del sistema cuando se somete a cargas extremas varias veces durante un corto periodo de tiempo.



* + Pruebas de resistencia (endurance): El fin de estas pruebas es comprobar si el sistema tiene la capacidad de soportar una carga de forma continua.



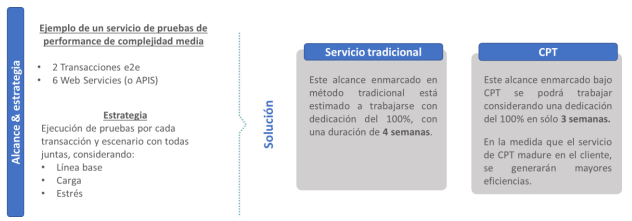
* Pruebas de disponibilidad:
  + Pruebas de larga duración o estabilidad (Stability test): Evaluar cómo se comporta el sistema bajo una carga constante de usuarios durante un período prolongado.
  + Pruebas de tolerancia a fallos o resiliencia (resilience test): Evaluar la capacidad del sistema para recuperarse de fallos inesperados en sus componentes.
* Otras pruebas:
  + Pruebas de volumetría (batch): son un tipo de prueba de software que se enfoca en la ejecución y validación de procesos que se ejecutan en lotes, es decir, una serie de trabajos o tareas que se agrupan y se ejecutan sin interacción humana directa.
  + Pruebas UX: APDEX (Application Performance Index o índice de rendimiento de aplicaciones): Aspectos de accesibilidad y disponibilidad, tiempos de espera.
  + Performance Tunning: Es un escenario colaborativo que involucra a las áreas de: Desarrollo de soluciones. Base de datos. Monitoreo de la infraestructura. Arquitectura e infraestructura. Redes y comunicaciones. Desarrollo de software. Pruebas de performance. Operaciones o soporte productivo.​​​​​​​ Y realizan cambios al sistema de manera directa y en línea.
  + Performance express: Servicio de pruebas para un alcance acotado de una transacción, mostrar valor sobre una necesidad puntual.

1. Aspectos claves:

* Objetivos de la prueba: Motivaciones y expectativas por las cuales se pretende realizar la prueba de performance, riesgos y situaciones.
* Criterios de Aceptación: Conocer los criterios, bajo los cuales serán medidas las características de performance esperadas por el cliente, según sus riesgos, necesidades o expectativas.
  + Velocidad en las transacciones desde el punto de vista del usuario.
  + Número de usuarios nombrados.
  + Número esperado de usuarios por unidad de tiempo (Usuarios Normales).
  + Número esperado de transacciones por unidad de tiempo.
  + Número Pico de usuarios por unidad de tiempo (Usuarios Pico).
  + Duración máxima del pico (Duración Pico).
  + Nivel de Tolerancia en Tiempo de respuesta.
  + Máximo de usuarios x unidad de tiempo.
  + Carga esperada.
* Ambiente de la Prueba y sus características: Conocer el ambiente físico y lógico del entorno donde se realizarán las pruebas, y el entorno donde se hará uso real del aplicativo.
* Data: Toda prueba requiere datos asociados al alcance definido, en performance se necesita un volumen de datos que soporte las distintas pruebas a realizar.
* Monitoreo: contar con monitoreo de la plataforma tecnológica que permita correlacionar información frente a los escenarios de pruebas.

1. Modelo de servicio:

* Performance tradicional: Trabaja bajo un alcance especifico, se planean tareas secuenciales, la solución técnica debe estar en sobre un 80%, se ejecutan en el ciclo final de las pruebas, se recomienda usar un entorno parecido al de producción.
* Continuos performance testing: Bajo modelos de operación continua, cuyo fin es incrementar el feddback temprano, unir las pruebas con DevOps, incrementar la cobertura y autonomía de pruebas, efectividad y frecuencia de pruebas, ecosistema tecnológico para performance.
  + - Informe de operación y seguimiento
    - Pipelines de performance y scripts de pruebas.
    - Informe y análisis de errores.



1. Requisitos:

* Estabilidad desde el punto de vista funcional.
* Conexiones entre los componentes.
* Cohesión con otras aplicaciones o ambientes de prueba.

1. Etapas del proyecto:

* Pre-venta:
  + Contextualización y entendimiento de la necesidad.
  + Definir el alcance de las pruebas.
  + Requisitos definidos.
    - Propuesta de servicios.
    - Seguimiento de servicios.
* Planeación de las pruebas.
  + Datos de prueba.
  + Infraestructura de pruebas.
  + Criterios de aceptación.
  + Alcance y estrategia.
    - Plan de pruebas de Performance.
* Diseño de las pruebas:
  + Especificación de los escenarios de pruebas.
  + Escenarios de prueba.
  + Guiones de prueba.
  + Automatización de pruebas.
    - Script de pruebas.
* Ejecución de pruebas.
  + Configuración de ambiente.
  + Línea base: Consiste en ejecutar cada petición con un único usuario.
  + La ejecución se hace de forma gradual, usualmente de 20%, 40%, 60%, 80%.
  + El monitoreo se hace sobre los indicadores
    - Reporte de ejecuciones.
* Análisis de resultados.
  + Re-ejecución de pruebas, cuando se implementaron cambios al haberse encontrado oportunidades de mejora.
    - Informe de cierre o resultados: Se realiza una vez se haya alcanzado los objetivos, se detallan los resultados.

ESTADISTICA:

1. Definición: La estadística es la ciencia encargada de recopilar, organizar, procesar, analizar y interpretar datos para describir fenómenos y predecir acontecimientos.
2. Descriptiva: Organizar reunir y comunicar información numérica.
   * Medidas de tendencia central, resumir en un solo valor un conjunto de datos:
     + Media o promedio: Valor promedio.
     + Mediana: Posición intermedia de los valores jerárquicamente.
     + Moda: Valor que más se repite.
   * Medidas de dispersión, que tanto se alejan los datos de la media aritmética.
     + Varianza: Es una medida estadística que indica qué tan dispersos están los datos de un conjunto respecto a su media.
     + Desviación estándar: Raíz cuadrada de la varianza, oscilan entre los datos determinados datos positivo y negativo.
     + Rango: El valor máximo del mínimo, el límite de la información.
   * Medidas de forma o distribución, como se juntan los valores de una muestra determinada:
     + Campana de Gauss: Identificación de tendencias, control de procesos.
       - Simetría: Normal
       - Asimétrica: Positivo mayores datos en el cuadrante 2 / Negativo mayor en el 1
3. Inferencial: Conduce a conclusiones de población para inferir datos de un grupo particular.
4. Datos:
   * Categóricos o cualitativos: Nominales / Ordinales
   * Numéricos o cuantitativos: Discretos y continuos.
5. Percentiles: Los percentiles son medidas estadísticas que indican cómo se distribuyen los tiempos de respuesta de las muestras de una prueba. Los percentiles representan el porcentaje de muestras cuyos tiempos de respuesta son iguales o inferiores a un valor dado.