**Clase 12. Pruebas No funcionales**

# Bases de datos & verificación de resultados

## Base de datos(DB)

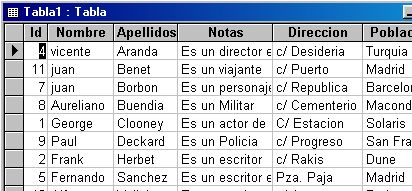
### ¿Qué es una base de datos?

Los datos por sí solos son cifras, letras, palabras o cualquier otro tipo de formato que se ingresa en una computadora para posteriormente ser procesada y utilizarla como información para tomar decisiones. Las bases de datos son el lugar donde estos datos se alojan para su posterior uso.

### Componentes de una base de datos

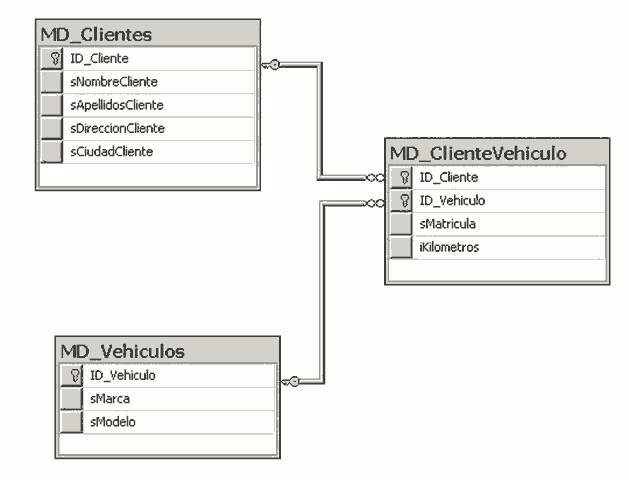
Una base de datos está compuesta por tablas que se relacionan entre sí, que contienen datos en celdas, que se alojan en la intersección de filas y columnas. La forma más gráfica que tenemos para entender una, es mirando una planilla de excel.

### Ejemplo

Podemos observar como las columnas nos marcan qué datos va a guardar la celda, y las filas nos indican el orden con el ID.

Fuente: [ResearchGate](https://www.researchgate.net/figure/Figura-44-Ejemplo-de-base-de-datos-de-comunidad-de-peces_fig5_290982379)

Ejemplo

Podemos observar cómo se relacionan las bases de datos.

### Tipos de bases

Existen muchos tipos de bases de datos, ya que con el tiempo fueron cambiando. Originalmente solo guardaban datos como tipos de formato, pero las mismas empezaron a crecer y hoy en día pueden contener:

* Imágenes
* Documentos
* Gráficos
* Audio
* etc...

### Creación de una base de datos

Generalmente, la creación y manejo de la base de datos, es realizada por los desarrolladores como una necesidad de guardar datos para poder utilizar en el proyecto. Pero existe un profesional que maneja bases de datos a gran escala llamado Database Administrator (también conocido como DBA), que será el experto en ellas, y podrá mejorar su relación y tiempo de respuesta.

### Lenguaje SQL

Las bases de datos necesitan su propio lenguaje para ser consultados, y para esto existe SQL (Structured Query Language) que es la forma en la cual podemos escribir para obtener algún dato que deseemos.

#### Ejemplo

SELECT hace referencia a que quiere buscar los “first\_name”, FROM indica que tabla que es la llamada “employees” y finalmente WHERE indica que solo traiga cuando el campo YEAR sea igual a 2000.

Por ende esta consulta traerá el primer nombre de los empleados de la tabla del mismo nombre que hayan sido contratados en el 2000.

Fuente[: SQLTutorial](https://www.sqltutorial.org/sql-syntax/)

#### Video ejemplo

Fuente:[enlace](http://www.youtube.com/watch?v=yLoh2sSDECw)

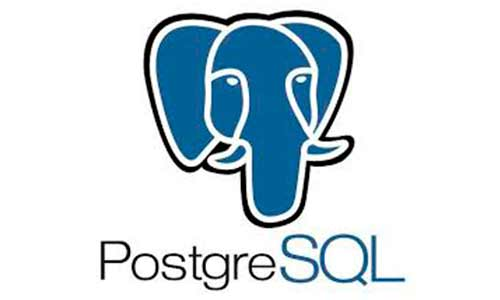
* Qué es sql 0.42 a 1.26
* Qué es base de datos 1.28 a 2.19
* SGBD(sistema de gestión de base de datos) 2.21 a 3.15
* Base de dato relacional 3.17 a 4.36
* DDL DML 4.37 a 5.24
* Ejemplo sentencias 5.25 a 10.28

### Motores de base de datos

Es necesario un programa que pueda administrar las bases de datos y genere la visibilidad de los datos, y a esto se los llama motores (o gestores) de bases de datos.

Existen muchos programas hoy en día, pero les mostraremos algunos de los más importantes en la actualidad.

#### Ejemplo



Fuente[: RevistaDigital](https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/)

### ¿Testeando base de datos?

Las pruebas de testers sobre bases de datos en sí, son prácticamente nulas, ya que no es un área que esté dentro de nuestros límites de trabajo. Aunque pueden solicitarnos que veamos las tablas, sus columnas y filas. Nuestra responsabilidad es la verificación de los datos existentes.

## Verificación de datos

### Uniendo temas

Con lo aprendido en la clase pasada sobre APIs, podemos comprender la verdadera trama de nuestra responsabilidad con las bases de datos. La verificación de los datos, contra lo obtenido en llamadas de APIs.

### Verificación

Con las pruebas de API que vimos en la clase pasada, pudimos aprender a asegurarnos que funcionen, pero sabiendo que es lo que está del otro lado, podemos asegurarnos de que los datos recibidos sean verídicos y lograr una verificación de los mismos

### Comparando valores

Al momento de querer comparar valores, sin conocer sobre SQL y Base de Datos, podemos pedir un extracto de la base para tener de referencia la información que estamos buscando. De esta forma, al ingresar parámetros en llamados de API, podemos comparar que los resultados obtenidos están bien en correlación con la base.

### Buscando integridad

Nuestra responsabilidad como testers contra las bases de datos, es poder asegurar la integridad de las mismas, para que todas las consultas y llamadas sean como tienen que ser.

Una base de datos sin integridad, no puede cumplir su función.

### Video ejemplo

Fuente:[enlace](http://www.youtube.com/watch?v=i08BQCfEbZI)

Aca podemos ver como se hizo un llamado con Postman, y posteriormente desde la consola verificó que esos sean los datos en la base de datos.

* Inicia BD 0.17 a 0.37
* Inicio aplicación 0.38 a 1.10
* Uso de postman 1.11 a 5.00
* Prueba con postman 5.00 a 6.23
* Prueba 6.23 a 7.45

# Pruebas de estrés

## ¿Qué buscamos?

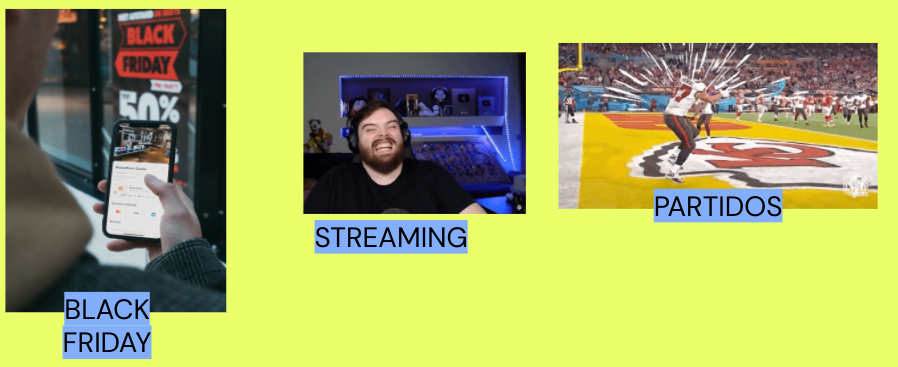
Una de las patas de la calidad, es poder asegurar la estabilidad y salud de un ambiente productivo del sistema. Las pruebas de estrés buscan saber hasta cuantas conexiones en simultáneo, por ejemplo, puede soportar un sistema sin romperse y explotar por los aires, imaginariamente hablando

## Picos de conectividad

Para entender un poco más en profundidad esto, debemos saber que los equipos físicos de los sistemas, tienen límites en todos sentidos, ya sea servidores, bases de datos, computadoras virtuales, etc.

Las empresas suelen conocer la cantidad media de conexiones al sitio, pero existen circunstancias o momentos donde los mismos sufren picos, y tenemos que asegurarnos que el sistema siga funcionando, incluso en esos momentos.

### Ejemplos picos de conectividad



Servicios de streaming con partidos, o páginas de compras en cybermonday por ejemplo.

### Mediciones

Las pruebas de estrés miden la capacidad de la infraestructura, los tiempos de respuesta de la plataforma en base a la cantidad de usuarios que están interactuando con el sistema y, con estas mediciones, se conocerá cual es el pico de interacciones que soporta el sistema antes de que se genere una negación de servicio.

## Estresando un sistema

Una prueba de estrés tiene como función medir cuanto un sistema puede soportar antes de que se sature o empiece a fallar.

## Pruebas esenciales

Existen muchos tipos de pruebas de estrés, como dijimos varias veces, tantas como te imagines. Es por eso, que aquí hablaremos de las 4 más importantes:

* LOAD TESTING
* STRESS
* ENDURANCE
* SCALABILITY

### Load Testing

#### Pruebas de carga

La prueba de carga nos permite saber la cantidad de peticiones que un sistema puede soportar antes de deshabilitarse.

Por ejemplo, se puede probar un programa para identificar si se cumple con la cantidad de peticiones concurrentes que se establece en un inicio.

#### Carga Landing

Una primera prueba sencilla, es las pruebas de carga de página. Nos indica cuanto tiempo tarda, y que las hace tardar. Estas herramientas son sencillas y muchas veces son algo muy sencillo de usar.

### stress

#### Pruebas de estrés

Una prueba de estrés consiste en probar los límites que un sistema puede soportar.

En este tipo de pruebas se busca enviar más peticiones de las que el software está acostumbrado a manejar para saber el comportamiento de la aplicación.

### Endurance

#### Pruebas de resistencia

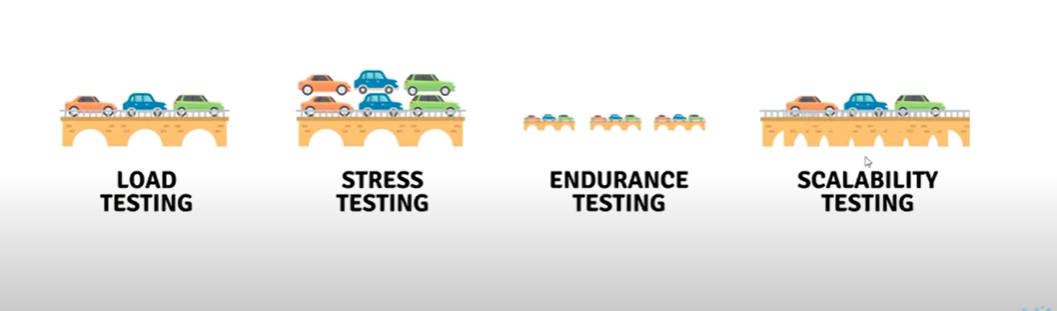
La prueba de resistencia se hace enviando peticiones a un sistema en ciertos intervalos de tiempo. De esta forma busca ver si a pesar de los picos de conexiones, el sistema sigue funcionando igual o cambia. Estos picos, no van a ser como en las pruebas previas, pero si busca medir la estabilidad del sistema.

### Scalability

#### Pruebas de escalabilidad

Las pruebas de escalabilidad es un factor que al evaluar nos permite identificar qué cosas se pueden mejorar en la infraestructura donde se encuentra nuestra aplicación. Con esta prueba realizada, y los cambios aplicados, se pueden volver a generar las pruebas previas para ir escalando el servicio y ver como aumenta la performance del sistema.

#### Pruebas de performance



En la primera imágen, cargamos el puente con autos para saber cuanto soporta.

En la segunda imágen estresamos al puente con más autos de los que puede soportar para saber cual es su límite

En la tercera imágen, representa los intervalos de tiempo (por eso está tres veces el puente)

En la cuarta imágen, mejoramos la infraestructura del puente para que pueda soportar más

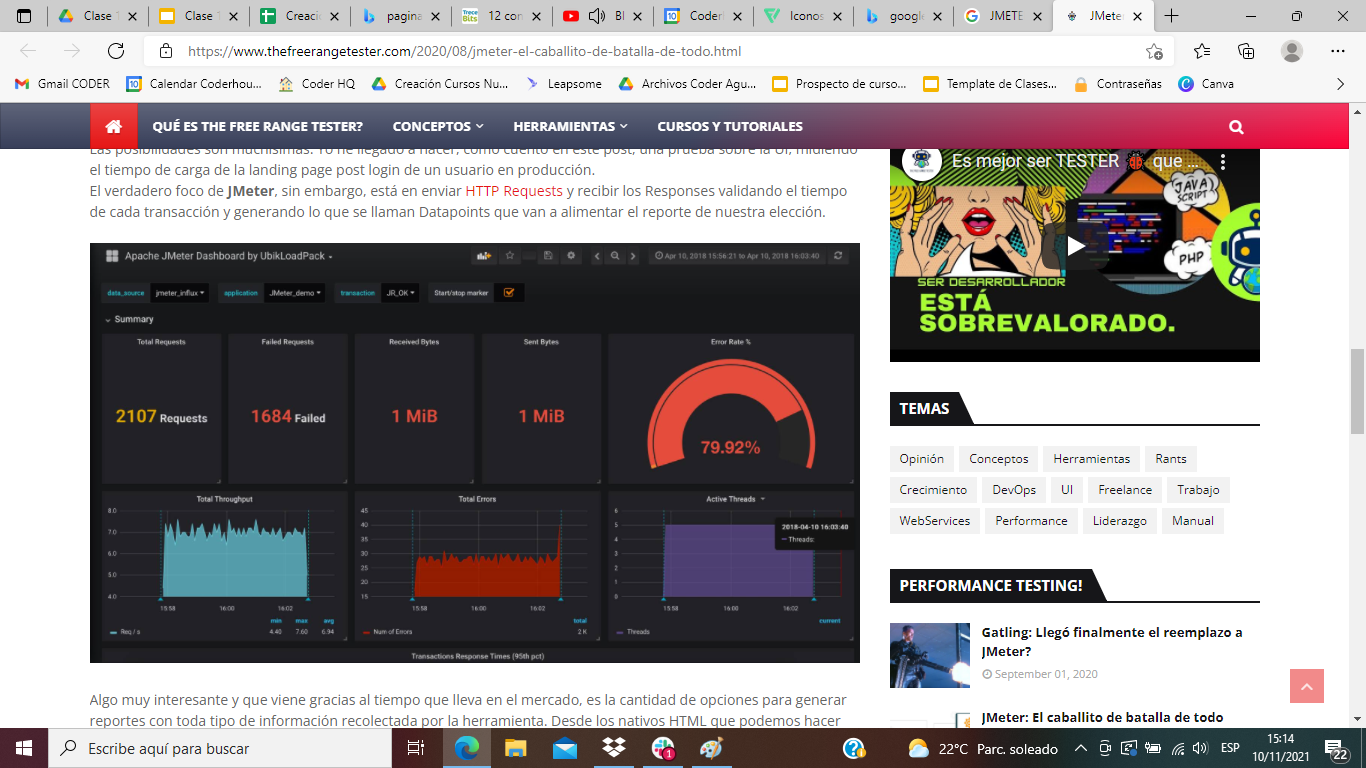
#### Otras pruebas

Recuerden que existen otros tipos de formas de estresar un sistema y eso es algo para investigar. Se puede estresar la pantalla de login o un sistema de búsqueda. También podemos medir tiempos de carga, no simplemente por estrés, sino como parte de las pruebas que llevan esto.

# Herramientas de estrés

## JMeter

JMeter es la herramienta más conocida para todo lo relacionado con pruebas de estrés. Esta herramienta contiene todo lo necesario para cualquier tipo de performance que se quiera medir de un sistema.



### Ejemplo JMeter

Fuente: [Enlace](http://www.youtube.com/watch?v=9cW1ud35xZMEn%20este%20video%20podremos%20observar%20dos%20tipos%20de%20pruebas%20de%20estres%20con%20la%20herramienta%20de%20forma%20sencilla.)

En este video podremos observar dos tipos de pruebas de estrés con la herramienta de forma sencilla.

## Presentando resultados

### Resultados positivos

Las métricas como resultados se utilizan de dos formas. Frente a respuestas positivas, el sistema necesitará una revisión en una temporalidad elegida por el equipo técnico para saber que esos números se mantienen en el tiempo.

O deberá corregirse algo.

### Resultados negativos

Cuando las métricas traen resultados malos, se hará el informe y la responsabilidad deberá ser llevada entre dos o cuatro áreas distintas. En un primer lugar, los desarrolladores deben trabajar con los sysadmins o infraestructura para ver el problema, y esto podría escalar a un arquitecto de software o a un database administrator, si el problema es una base de datos.

# Pruebas de seguridad

### Pentesting

Las pruebas de penetración, que en inglés son penetration testing, son las pruebas que QA debería hacer sobre la aplicación para evitar posibles brechas de seguridad.

### Seguridad en desarrollo

Es una tendencia en el mundo del desarrollo, dejar de lado la seguridad de las aplicaciones en manos de gente externa, pero es importante tener en cuenta que las primeras pruebas de seguridad tiene que ser generadas en el ciclo de desarrollo para evitar mal uso de los usuarios, ataques de seguridad, mala de calidad de software, etc...

### Seguridad en testing

Los aspectos de seguridad de información de las aplicaciones web deben ser responsabilidad de los probadores y tenerse en cuenta dentro de un proceso de calidad, porque también es un punto crítico viéndose involucrada información de bases de datos y la disponibilidad de la aplicación, para destacar algunas áreas.

### Necesidad de seguridad

Con esto se ve la necesidad de integrar la seguridad con el concepto de calidad del software, ampliando el concepto de pruebas de software y cómo se puede establecer un modelo que permita llevar a cabo esta difícil tarea de encontrar las vulnerabilidades de una plataforma web.

### Seguridad en testing

Los aspectos de seguridad de información de las aplicaciones web deben ser responsabilidad de los probadores y tenerse en cuenta dentro de un proceso de calidad, porque también es un punto crítico viéndose involucrada información de bases de datos y la disponibilidad de la aplicación, para destacar algunas áreas.

## Vulnerabilidades

### ¿Qué son las vulnerabilidades?

Las vulnerabilidades son todas las brechas de seguridad en un aplicación desde donde alguien puede aprovecharse para romper la aplicación o hacer algún robo de información. Ahora hablaremos de algunas de las vulnerabilidades más comunes y que desde nuestro lado podemos investigar.

### Autentificación

Consiste en la verificación de la identidad de un usuario que tiene permisos para acceder a la aplicación. Los atacantes usualmente buscan quebrantar cualquier mecanismo de autenticación que se les presente con el fin de ingresar a la información que requieren.

Fuente: [Enlace](https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf)

### Gestión de la configuración

El conocimiento de la arquitectura de la aplicación web puede revelar información valiosa para un atacante, como los protocolos utilizados, las funcionalidades administrativas, los métodos de cifrado utilizados o el servidor de aplicaciones utilizado que facilitan la labor de un atacante.

Fuente: [Enlace](https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf)

### Sesiones

La gestión de sesiones en una aplicación web es un aspecto importante en la seguridad de información, puesto que estas necesitan mantener el estado de las peticiones del cliente y a partir de las sesiones se pueden asociar entre sí; los navegadores almacenan información sobre las sesiones en cookies.

Fuente: [Enlace](https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf)

### Validación de datos

Tal como su nombre lo dice, se refiere a la verificación de datos de entrada permitidos dentro de la aplicación, y es una de las debilidades más frecuentes en las aplicaciones web.

Es importante partir de la premisa de que ningún dato ingresado es confiable; además, se debe definir qué se permite teniendo en cuenta el modelo de datos, la lógica del negocio y la definición de tipo de variables en la programación.

Fuente: [Enlace](https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf)

Las siguientes son los principales ataques de validación de datos:

Cross Site Scripting busca manipular las entradas al sistema para robar información confidencial.

Fuente: [Enlace](https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf)

Inyección SQL: se pretende introducir una consulta SQL cuando se permite el ingreso de caracteres propios del lenguaje SQL y que el motor de base de datos interpreta.

Desbordamiento de buffer: se pretende comprometer la integridad de la aplicación por inconsistencia entre los formatos, tipos y longitudes entre datos ingresados.

Fuente: [Enlace](https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf)

### Denegación de servicio

El principal objetivo de este tipo de ataque es saturar al servidor de la aplicación de tráfico de red, que a pesar de que las buenas prácticas de desarrollo no tienen mucho alcance en la tarea de evitarlo, sí es importante construir una arquitectura de software robusta.

Fuente: [Enlace](https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf)

## Límites de seguridad en testing

### Nuestro alcance

La seguridad en un desarrollo de aplicación deberá ser nuestra responsabilidad, pero hay que saber que el límite termina en el código y su funcionamiento. Hay otras formas de penetrar sistemas, que están fuera de nuestro alcance de conocimiento.

### ¿Quién está en las sombras?

Las empresas tienen sus propios equipos de seguridad, que buscan frenar ataques externos, a ellos se los conoce como Cyber Defense o Ethical Hackers (Hackers de guante blanco). Ellos son los responsables de ataques de gran escala, con herramientas mucho más avanzadas para trabajar problemas relacionados con terminales, servidores, bases de datos y otras áreas del sistema.

# Pruebas automatizadas

## Introducción

La automatización de pruebas se vuelve día a día un requerimiento cada vez más importante para el QA manual, debido al ahorro de tiempo que se gana al realizar esta actividad. Es por eso, que haremos un último estudio de esto.

## ¿Qué es la automatización de pruebas?

### Automatizando

La automatización es una herramienta que le permite al QA adelantar tiempos de pruebas ya hechos, a través de la programación de comportamiento de la aplicación.

La idea de esto, es asegurarnos que las funcionalidades funcionen como deben, y que ciertas secciones se encuentren en donde deben.

### Falsos conceptos

* Se puede automatizar todo:¡No! Sabemos que una aplicación no se puede probar a fondo nunca. Por ende, nunca se puede automatizar todo tampoco.
* Las pruebas automatizadas son demasiado costosas: Si y no. En una primera instancia, la inversión es alta, pero el mismo con el tiempo se beneficia y su costo baja mucho.
* Las pruebas automáticas son fáciles: ¡No! A pesar de que existan herramientas que hagan la labor más sencilla, automatizar es una de las partes más complejas del ciclo de desarrollo.
* Automatización reemplazará a pruebas manuales: No. Es el error más grande de todos. La automatización ayuda a lo manual, pero nunca podrá reemplazarlo.

### Tipos de pruebas de automatización

#### 3 Tipos

A grande escala, existen tres tipos de automatizaciones:

* Las pruebas unitarias.
* Las pruebas de API.
* Las pruebas de interfaz gráfica.

##### Pruebas unitarias automatizadas

Las pruebas unitarias generalmente son realizadas por programadores. Estas pruebas se escriben para probar el nivel de código.Existen numerosos frameworks para la automatización de pruebas unitarias. Algunos de los más conocidos son NUnit, Junit, Unit, js, QUnit, etc.



##### Pruebas automatizadas de APIs

Con las pruebas automatizadas de API, se prueban la funcionalidad, el cumplimiento y los problemas de seguridad, y si está probando una aplicación web, también se examina la seguridad de Solicitud y Respuesta. Para las pruebas de API automatizadas, puede usar LambdaTest que está disponible en versiones gratuitas y de pago.

##### Automatización de prueba gráfica

Las pruebas de automatización de la GUI suplantan las mismas acciones que los usuarios realizan con su mouse y teclado en la interfaz de usuario.

Existen múltiples herramientas de prueba de GUI, como Selenium,que es una de las más conocidas.

## Glosario

**Base de Datos:** es una recopilación organizada de información o datos estructurados, que se almacena de forma electrónica en un sistema informático

**SQL:** Lenguaje de Consulta Estructurada (Structured Query Language), es un lenguaje de programación diseñado para actualizar, obtener, y calcular información en bases de datos relacionales.

**JMeter:** es una herramienta que facilita la gestión integral de los procesos de pruebas de rendimiento

**Scalability Testing:** pruebas de escalabilidad

**Endurance Testing:** Pruebas de resistencia

**Pentesting:** El pentesting o test de penetración consiste en atacar diferentes entornos o sistemas con el objetivo de detectar y prevenir posibles fallos

**Automatización de pruebas:** es una práctica que permite controlar la ejecución de un producto software de manera automática, comparando los resultados obtenidos con los resultados esperados

**Load Testing:** pruebas de carga

#### ¿QUIERES SABER MÁS? TE DEJAMOS MATERIAL AMPLIADO DE LA CLASE

* [Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software | S. M. Diaz Diaz](https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf)

#DEMOCRATIZANDOLAEDUCACIÓN