

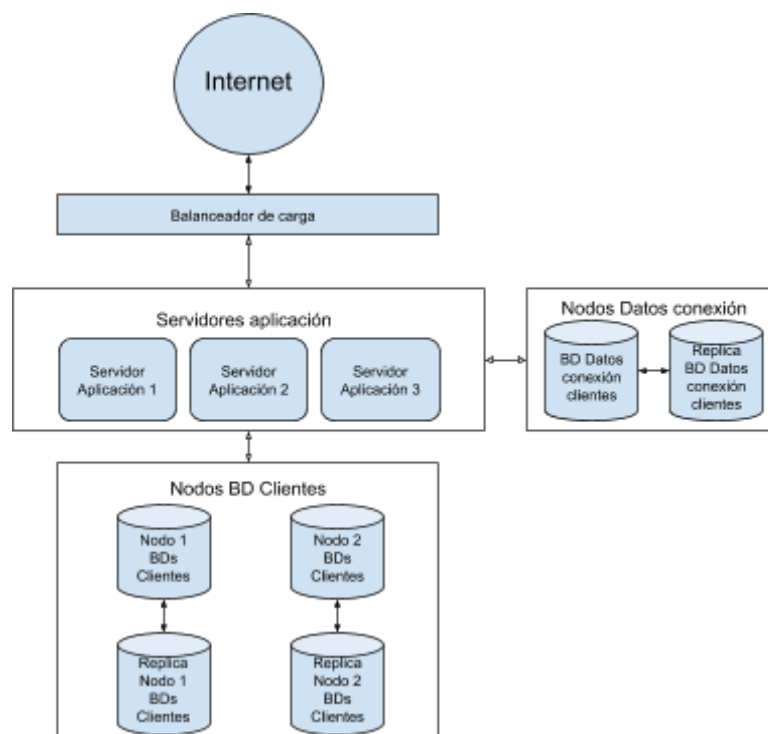
Healthatom Ingeniería

Ops:

1. Tenemos una empresa que ofrece un software para administración de talleres mecánicos como servicio (SaaS). Hoy en día cuentan con más de 1000 clientes en 10 países, cada uno con su propia base de datos y exactamente el mismo modelo de datos. Existe una diferencia máxima horaria de 4 horas entre todos los clientes (GMT -2 a GMT -6). Para poder deployar una nueva funcionalidad, se necesita correr una migración al modelo de datos que agrega 3 tablas con referencias a tablas existentes y además alterar una tabla grande existente (agregar índices) con un promedio por cliente de 10000 registros. Considerando este escenario, **¿qué precauciones se deben tomar para mitigar los riesgos de esta migración?**
2. La misma empresa del caso anterior tiene la siguiente infraestructura:
 - a. Tienen 3 servidores de aplicación ejecutando una aplicación PHP con nginx.
 - b. Frente a estos servidores, existe un balanceador de carga que distribuye las conexiones equitativamente.
 - c. Estos servidores de aplicación consultan los datos de conexión de cada cliente desde una base de datos maestra, para luego conectarse al nodo y base de datos correspondiente a cada cliente.
 - d. Existen dos nodos con bases de datos de cliente (cada uno con una replica de solo lectura) y un nodo maestro (con su replica) que entrega los datos de conexión. Todas las bases de datos usan MySQL.

Identifique todos los potenciales Single Point of Failure de esta infraestructura y además sugerencias para mitigarlos.

Modelo referencial:



Programación/Algoritmos:

1. Escriba una función/método en (cualquier lenguaje) que determine la cantidad de 0's a la derecha de $n!$ (factorial). Use para probar el número 1000.
2. Problema Etiquetadora defectuosa: En la empresa tenemos un juego de apuestas con dinero ficticio, en donde una vez se define el ganador, se le imprime un ticket con el monto ganado. Un día, al momento de querer imprimir, nos dimos cuenta de que el dígito 4 no se imprimía. Esto no es algo bueno, dado que siempre el monto es un número que contiene al menos un 4 entre sus dígitos. Por suerte, a una persona se le ocurrió que podíamos entregar dos cheques con montos A y B, tal que $A+B = N$, donde N es el monto total del ticket a pagar. Se necesita entonces formular una función que nos permita imprimir A y B de forma que podamos pagar al ganador.
 - a. Input: Monto N a pagar a la persona.
 - b. Output: Valores A y B tal que $A+B = N$
 - c. Restricciones: $1 \leq N \leq 10^{100}$. Al menos un dígito de N es 4
 - d. Ejemplo:
Entrada: $N = 9463$
Salida: $A = 6352, B=3111$

SQL:

Tenemos un modelo en MySQL (healthatom_test.sql) que refleja una pequeña clínica dental. Algunos detalles importantes del funcionamiento de esta clínica son los siguientes:

- En la clínica los profesionales generan presupuestos, los cuales a su vez pueden tener varios detalles con prestaciones cargadas.
- El precio de las prestaciones cargado a los presupuestos puede ser distinto al precio referencial de la prestación.
- Cada presupuesto tiene un dueño, pero cualquier profesional puede cargar más detalles o realizar los detalles de un presupuesto.
- Un presupuesto se considera finalizado cuando todos sus detalles están realizados.
- Un presupuesto se encuentra activo si tiene al menos una prestación sin realizar cargada. Presupuestos sin prestaciones no están activos.
- Un detalle se considera con deuda cuando está realizado.
- Un paciente se considera atendido en una sucursal cuando tiene presupuestos en dicha sucursal.

Necesitamos consultas para obtener la siguiente información del modelo (enviar respuestas ordenadas en archivo con nombre "consultas.sql"):

1. Totales adeudados por paciente:
 - Listado de todos los pacientes con su monto de deuda
 - Columnas: [id_paciente, nombre_paciente, deuda]
2. Totales realizados y por realizar por presupuesto (plata):
 - Listado de todos los presupuestos, especificando el total realizado y pendiente por separado
 - Columnas: [id_presupuesto, nombre_presupuesto, total_realizado, total_pendiente]

3. Cantidad de presupuestos activos por profesional:
 - Listado de profesionales con la cuenta de presupuestos activos de cada uno
 - Columnas: [id_profesional, nombre_profesional, total_presupuestos_activos]
4. Edad promedio pacientes por sexo:
 - Listado de todos los sexos con su edad promedio
 - Columnas: [sexo, edad_promedio]
5. Profesionales sin ninguna realización:
 - Listado de todos los profesionales que no tengan ninguna realización de detalle
 - Columnas: [id_profesional, nombre_profesional]
6. Totales realizados por profesional/categoría (plata):
 - Listado de totales realizados por profesional/categoría
 - Columnas: [id_profesional, nombre_profesional, id_categoria, nombre_categoria, total_realizado]
7. Presupuestos finalizados:
 - Listado de todos los presupuestos finalizados (sin prestaciones sin realizar ni vacíos)
 - Columnas: [id_presupuesto, nombre_presupuesto]
8. Paciente con mayor deuda:
 - Paciente con la mayor deuda total de la clínica
 - Columnas: [id_paciente, nombre_paciente, deuda_total]
9. Presupuestos sin prestaciones cargadas:
 - Listado de todos los presupuestos que no tienen prestaciones cargadas
 - Columnas: [id_presupuesto, nombre_presupuesto]
10. Tiempo promedio (en días) entre carga y realización de una prestación:
 - Promedio global de tiempo entre carga y realización de prestaciones en la clínica
 - Columnas: [tiempo_promedio_en_dias]
11. Especialidad profesional con más realizaciones:
 - Especialidad con mayor número de realizaciones totales entre todos los profesionales
 - Columnas: [id_especialidad, nombre_especialidad, total_realizaciones]
12. Precios promedio cobrados a pacientes por categoría:
 - Listado de categorías con precio promedio de sus prestaciones cargadas a presupuestos
 - Columnas: [id_categoria, nombre_categoria, precio_promedio]
13. Recaudación por categoría (prestaciones realizadas):
 - Listado de categorías con totales realizados de sus prestaciones (plata)
 - Columnas: [id_categoria, nombre_categoria, total_realizado]
14. Cantidad de presupuestos de pacientes con más de 1 presupuesto:
 - Listado de pacientes que cumplan la condición de tener más de un presupuesto y el total de presupuestos que tienen
 - Columnas: [id_paciente, nombre_paciente, total_presupuestos]
15. Pacientes sin deuda:
 - Listado de pacientes que no tienen deuda
 - Columnas: [id_paciente, nombre_paciente]

16. Cantidad de pacientes atendidos por sucursal:

- Listado de sucursales con el total de pacientes atendidos en cada una
- Columnas: [id_sucursal, nombre_sucursal, total_pacientes_atendidos]