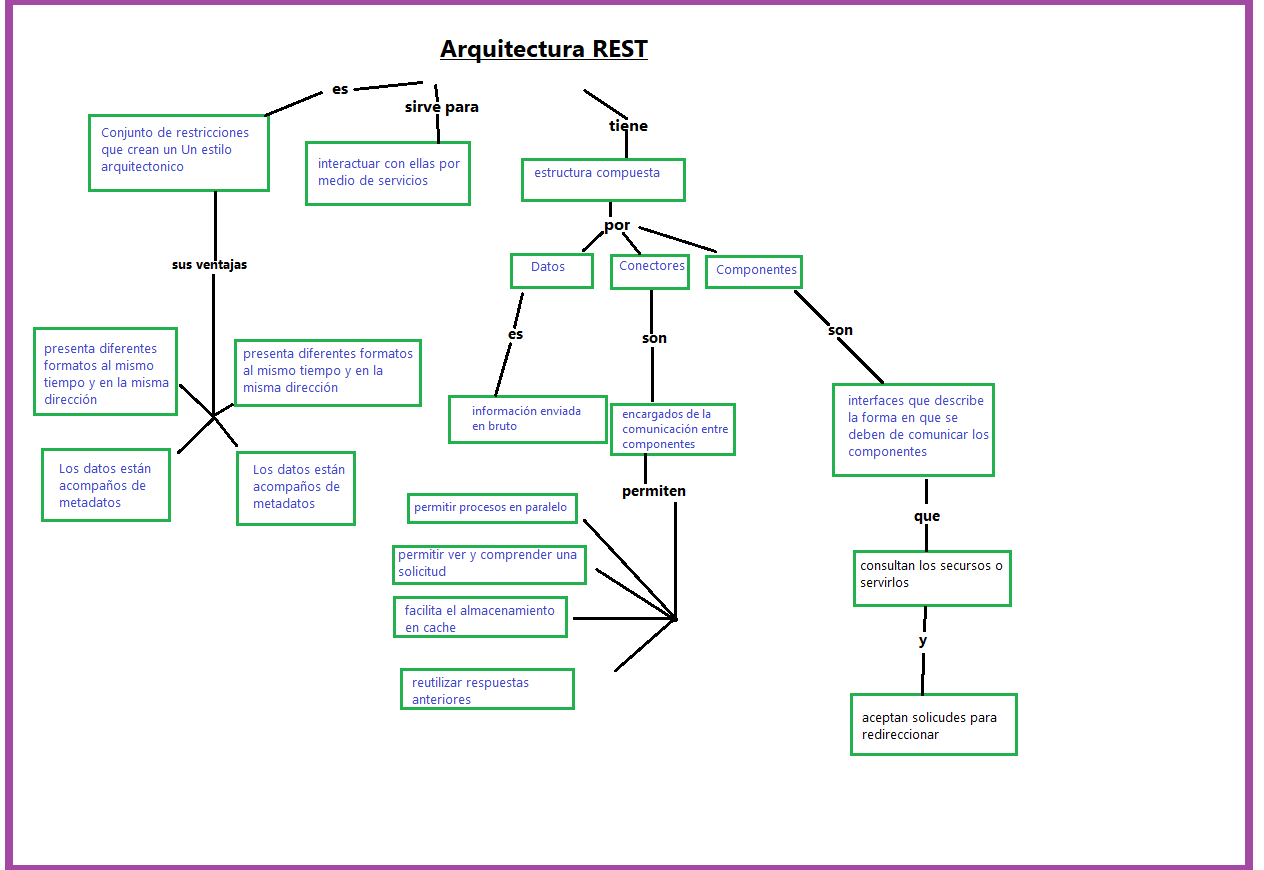
Presentado por

Maik Poveda Barbosa

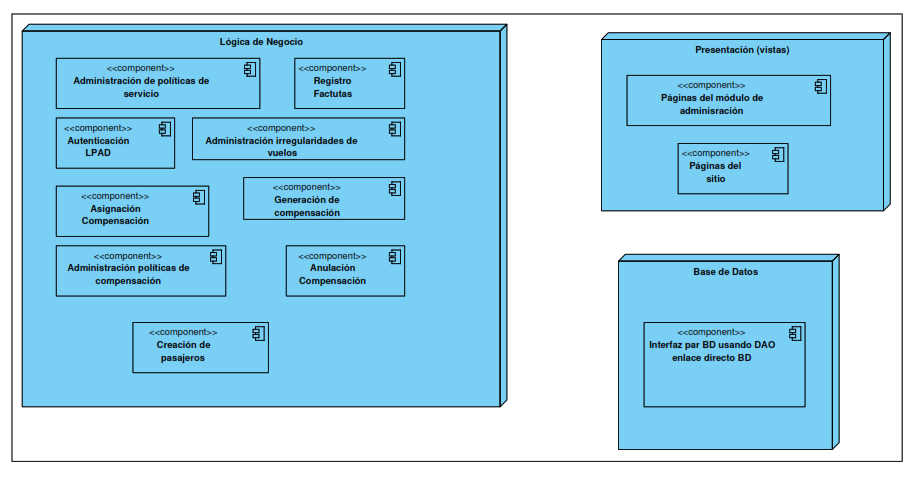
Leonardo Aponte

**Arquitectura REST**

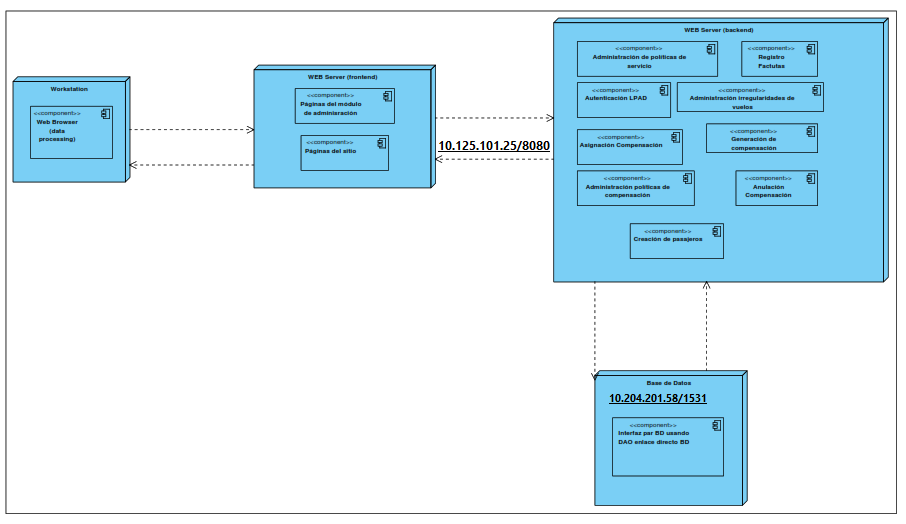
**Mapa conceptual**

****

**Diagrama de Componentes**

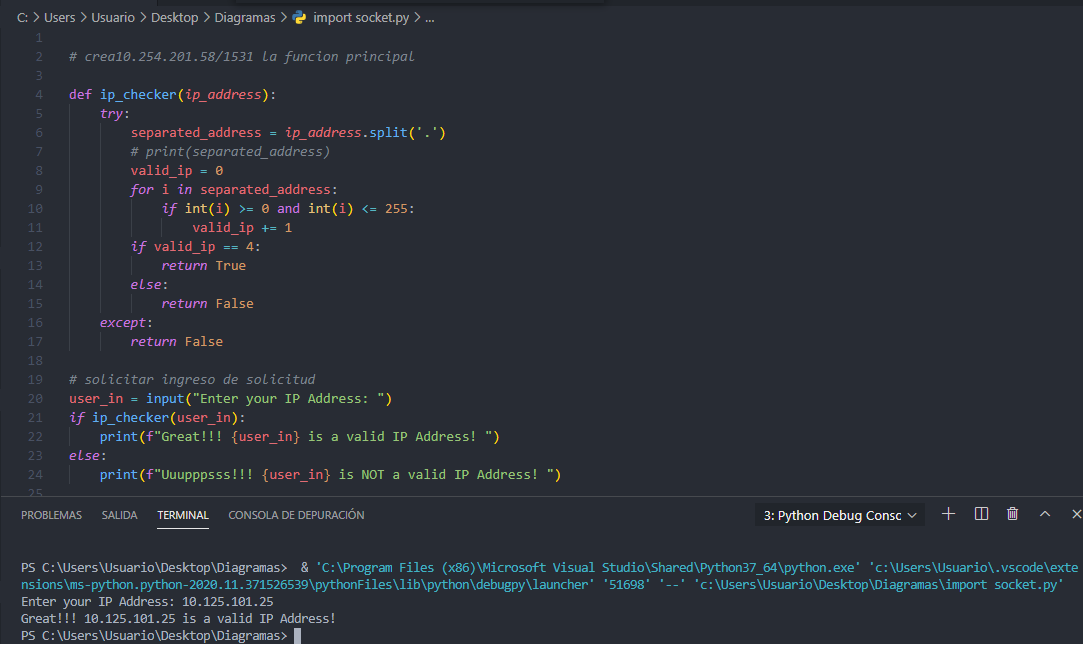
****

**Diagrama de Despliegue**

****

**Validación IPv4**

****

****

**EJERCICIO**

Suponer que el backend (aplicación) está en un solo servidor.

1. ¿Como realizo escalamiento vertical de una aplicación basada en REST? ¿Que necesito para poderlo realizar?

Para este escalamiento, la ampliacion de nodos que permita

para que amplie a un nodo, y los cluster tenga dos servidores con las estancias requeridas. Se amplía los cluster para que se realice el proceso

<Basico dos servidores REST, un balanceador de carga>

2. ¿Como realizo escalamiento horizontal de una aplicación basada en REST? ¿Que necesito para poderlo realizar?

Se amplía los recursos del contenedor para que responda más, o migrar o escalar el servidor o se realiza escalamiento con load balancer.

Este modelo implica tener varios servidores (conocidos como Nodos) trabajando como un todo. Se crea una red de servidores conocida como Cluster, con la finalidad de repartirse el trabajo entre todos nodos del cluster, cuando el performance del cluster se ve afectada con el incremento de usuarios, se añaden nuevos nodos al cluster, de esta forma a medida que es requeridos, más y más nodos son agregados al cluster.

3. ¿Puedo utilizar un load balancer en una aplicación basada en REST? Defina su respuesta.

Se implementa un balanceador de carga de dos servidores rest, Para su correcto funcionamiento deberá existir un servidor primario desde el cual se administra el cluster. Cada servidor del cluster deberá tener un software que permite integrase al cluster, para que cuando llegue las peticiones HTTP, distribuya al servicio que se requiera, por ejemplo, para las aplicaciones Java, tenemos los servidores de aplicaciones como Weblogic, Widfly, Websphere, etc. y sobre estos se montan las aplicaciones que queremos escalar.

4. ¿Puedo implementar algún estilo arquitectónico visto con REST? ¿Cuál y por qué?

Se puede implementar, relizando buenas prácticas de Desarrollo con Patrón de capas

Este patrón se puede utilizar para estructurar programas que se pueden descomponer en grupos de subtareas, cada una de las cuales se encuentra en un nivel particular de abstracción. Cada capa proporciona servicios a la siguiente capa superior.

5. Como puedo utilizar todos los endpoint vistos en clase en un mismo servidor para que el backend tenga alta disponibilidad - Recordar que la disponibilidad es un atributo de calidad. (incluye explicación de la fórmula)

Comencemos definiendo La disponibilidad, es una de las características de las arquitecturas empresariales que mide el grado con el que los recursos del sistema están disponibles para su uso por el usuario final a lo largo de un tiempo dado. En este caso de estudio, se define el sistema que tiene un acuerdo de niveles de servicio que alcanzan las 24 horas al día, los 365 días del año. Así entonces, suponiendo un sistema con un SLA (Service Level Agreement – SLA), que defina cuánto tiempo y en qué horarios debe estar en línea, definiendo el servicio de 24×365 podríamos calcular su disponibilidad de la siguiente manera:

Disponibilidad = ((A – B) /A) x 100 por ciento)

Donde:

A = Horas comprometidas de disponibilidad: 24 x 365 = 8,760 Horas/año.

B = Número de horas fuera de línea (Horas de “caída del sistema” durante el tiempo de disponibilidad comprometido). Por ejemplo: 15 horas por falla en un disco; 9 horas por mantenimiento preventivo no planeado.

así entonces:

Disponibilidad = ((8,760 – 24) /8,760) x 100 por ciento) = 99.726%

Este sería un caso excepcional, ya que ese porcentaje es dificil de alcanzar, ya que depende de factores como:

* La severidad del problema.
* La disponibilidad del personal de soporte.
* La complejidad del sistema

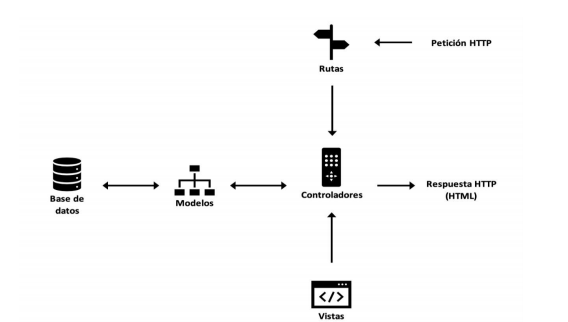
6. ¿Cuál sería la diferencia entre un servicio web SOAP y RESTful?

REST es un conjunto de pautas que ofrece una implementación flexible, mientras que SOAP es un protocolo con requisitos específicos, como en el caso de la mensajería XML.

los servicios web de RESTfull, formato preferido para los mensajes es la notación de objetos JavaScript (JSON), ya que, a pesar de su nombre, puede leerlo cualquier lenguaje de programación, es ligero y lo comprenden tanto las personas como las máquinas. De esta forma, las API de RESTful son más flexibles y se pueden configurar con mayor facilidad

El uso de REST frente a SOAP nos proporciona una mayor escalabilidad del servicio. REST proporciona una arquitectura mucho más fácil, así como peticiones más simplificadas, permitiendo todo ello un mayor rendimiento y velocidad; y permite interpretar las URI de manera visual. REST se ha convertido en la implementación más utilizada en la actualidad

7. ¿Como puedo consumir un servicio RESTfull en javascript, angular, vue o react?



• Si realizamos CRUD, debemos utilizar los HTTP verbs de forma adecuada para cuidar la semántica.

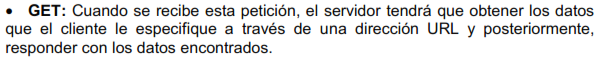
o GET: Obtener datos. Ej: GET /v1/empleados/1234

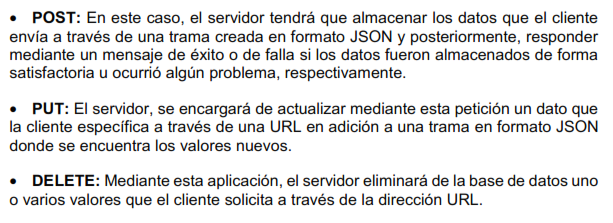
o PUT: Actualizar datos. Ej: PUT /v1/empleados/1234

o POST: Crear un nuevo recurso. Ej: POST /v1/empleados

o DELETE: Borrar el recurso. Ej: DELETE /v1/empleados/1234

o ¿PATCH?: Para actualizar ciertos datos





8. Como puedo consumir un servicio RESTfull en Android, Swift, react native, ionic o flutter

9. Realice el consumo del endpoint de cada grupo con los 4 métodos vistos en clase, realice una descripción del resultado de cada uno.

