



Facultad  
de Ingenierías

# Aplicaciones para la Nube

## Aplicaciones Multi-tenant

César Augusto Díaz Arriaga



# César Augusto Díaz Arriaga

Lider de Desarrollo GIA-UTP

- Ingeniero de Sistemas y Computación
- Estudiante de Maestria en Ingenieria de Sistemas y Computación.
- 20 años aprox. de experiencia en desarrollo de aplicaciones empresariales
- Docente Catedrático
- Apasionado por escribir código de calidad y de fácil entendimiento.
- Me gusta el Baloncesto, Volleyball y programar.
- Programador del eje cafetero que no toma café.



cesardiaz-dev



@cesardiaz\_dev



cesardiaz-dev  
cesardiaz-utp

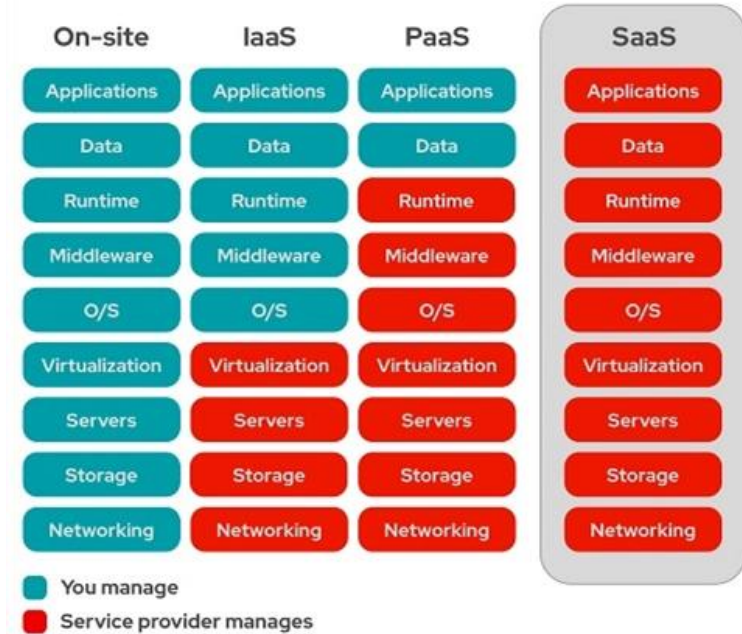


# Aplicaciones en la Nube

## Software as a Service - SaaS

Según Wikipedia:

**Software como un Servicio**, abreviado **ScuS<sup>1</sup>** (del *inglés*: **Software as a Service, SaaS**), es un modelo de distribución de *software* donde el soporte lógico y los datos que maneja se alojan en servidores de una compañía de *tecnologías de información y comunicación* (TIC), a los que se accede vía *Internet* desde un cliente. La empresa proveedora TIC se ocupa del servicio de mantenimiento, de la operación diaria y del soporte del software usado por el cliente. Regularmente el software puede ser consultado en cualquier computador, se encuentre presente en la empresa o no. Se deduce que la información, el procesamiento, los *insumos*, y los resultados de la *lógica de negocio* del software, están hospedados en la compañía de TIC.



# Ventajas del SaaS

- Obtener acceso a aplicaciones sofisticadas.
- Pagar sólo por lo que usa.
- Usar software de cliente gratuito.
- Movilizar fácilmente a su personal.
- Obtener acceso a los datos de las aplicaciones desde cualquier parte.

# ¿Y como desarrollamos aplicaciones SaaS?



¿Multi tenant?



# ¿Multitenencia?

Según Wikipedia:

***Tenencia múltiple*** o multitenencia en informática corresponde a un principio de [arquitectura de software](#) en la cual una sola instancia de la aplicación se ejecuta en el servidor, pero sirviendo a múltiples clientes u organizaciones (tenedor o instancia).



Una aplicación, ¿múltiples clientes?





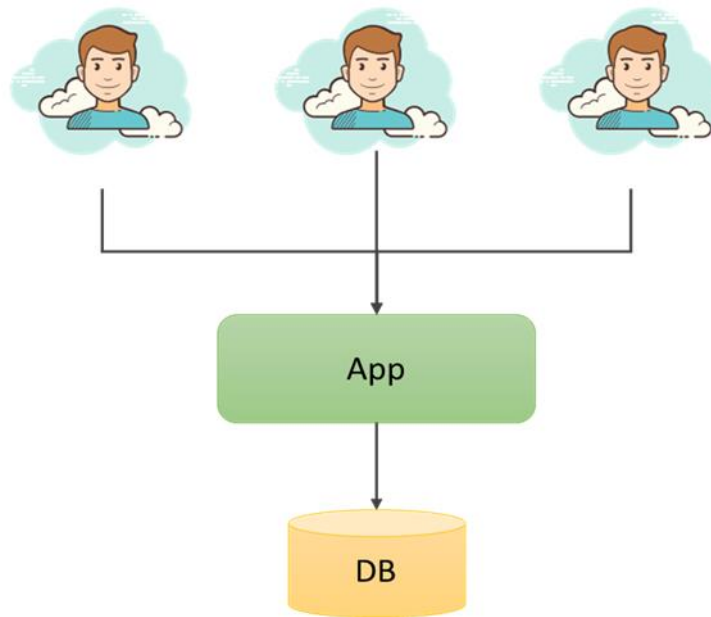
# Alternativas para Multi tenencia

1. Multi tenencia con base de datos y esquema compartido
2. Multi tenencia con base de datos compartida, diferente esquema
3. Multi tenencia con diferente base de datos

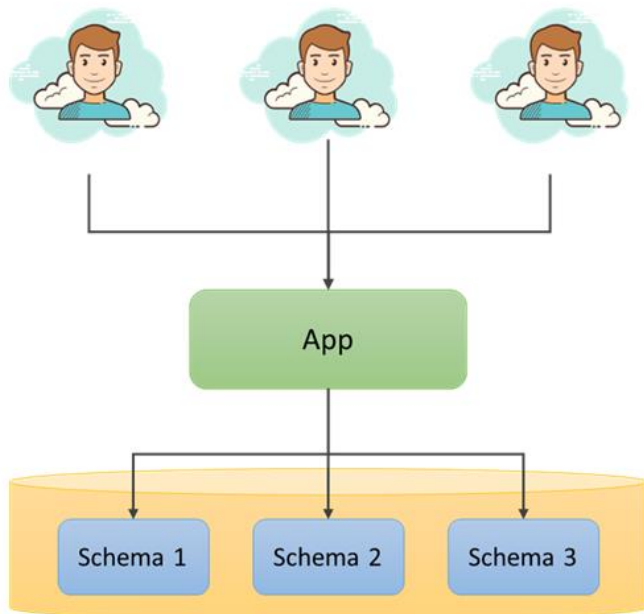
# 1. Base de datos y esquema compartida

Esta alternativa es la que se usa por defecto en la mayoría de aplicaciones (es la más fácil de implementar), ya que se separa los usuarios de la aplicación por una relación con la tabla de usuarios.

Este es la forma más simple de crear una aplicación Multi-Tenant, la cual consiste en compartir una misma base de datos y esquema para todos los usuarios registrados, diferenciando los datos de cada tenant por un identificador que pondremos en todas las tablas.



## 2. Base de datos compartida, diferente esquema



Cada cliente (tenant) tiene su propio esquema de BD

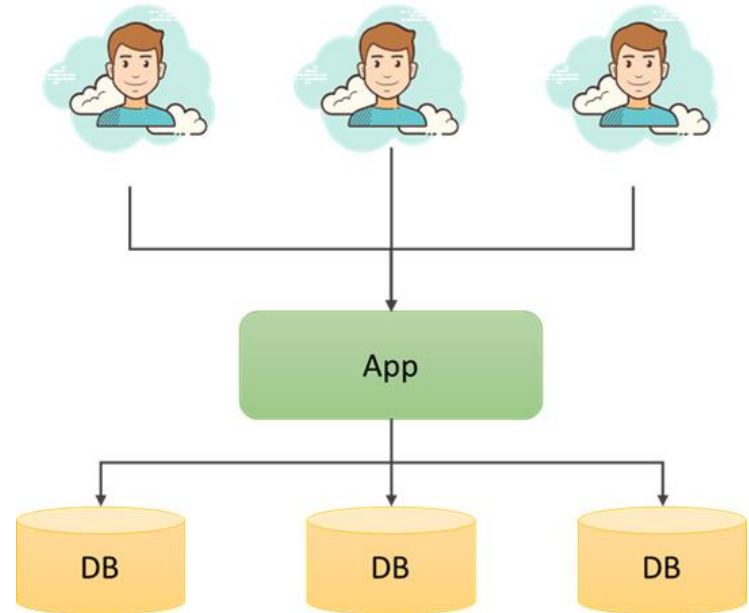
Este método es muy similar al anterior, ya que los clientes comparten la base de datos, con la única diferencia de que la información se almacena en esquemas diferentes

Esta alternativa permite usar cualquier motor de base de datos, se puede relacionar como el proceso de virtualización en el que cada instancia se implementa para el nuevo cliente.

### 3. Base de datos por inquilino

Cada instancia de base de datos está separada en un servidor totalmente separado, por lo que se pueden cumplir necesidades como que la información esté alojada en distintas regiones.

Esto implica que desde una organización no se puede acceder a los datos de otra, por lo que el nivel de aislamiento es alto e implica grandes ventajas.



y ahora, ¿cual elegimos?




# Aspectos claves de las aplicaciones Multi-Tenant

- El mecanismo para la distribución de recursos
- El mecanismo de aislamiento de seguridad
- El mecanismo de personalización



# DEMO

<https://github.com/cesardiaz-utp/2021-Multitenancy>



# Stack Tecnológico (Demo)

- **Lenguaje:** Java - OpenJDK 11
- **Framework:** Spring Boot 2.5.3
  - Spring Web
  - Spring Data JPA
- **Persistencia:** Hibernate 5.4.32
- **Migración:** Flyway 7.7.3
- **Base de datos:** H2 Embedded 1.4.200
- **Utilidades:**
  - Lombok 1.18.20



<https://github.com/cesardiaz-utp/2021-Multitenancy>





Universidad Tecnológica  
de Pereira

Facultad  
de Ingenierías

# GRACIAS

César Augusto Díaz Arriaga