



Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación

Mauricio Gaona
mauricio.gaona@correounivalle.edu.co

Profesor

2023-I

Desarrollo de Software I



01

RESUMEN

02

MODELO DE DESPLIEGUE

03

ARQUITECTURAS ÁGILES

04

ESTADO DEL PROYECTO

Resumen

Universidad
del Valle



Estimación de Tiempos en la metodologías ágiles

Resumen

Sprint	HU	Puntos de HU	Desarrollador
1	HU1	4	Pedro/María Juan/Ricardo
	HU2	4	
2	HU3	2	Maria Pedro Juan
	HU4	4	
	HU6	3	
3	HU5	4	Maria Juan
	HU7	4	
4	HU9	5	Pedro Juan
	HU8	3	
5	HU10	4	Maria Pedro
	HU11	5	
		Total puntos del proyecto: 35 Puntos promedio por Sprint: 7	

Estimación de esfuerzo (tamaño del software) Total Puntos del proyecto	(Product Backlog)	35 puntos
Velocidad en puntos del equipo por mes	Puntos por mes (2 Sprints)*	14 puntos
Duración estimada del proyecto	Puntos del proyecto / Velocidad en puntos del equipo por mes	2,5 meses

* Asumiendo sprints de 2 semanas

3 meses



Costo de una App en las metodologías ágiles

Resumen

	Costos del desarrollo		
PB	Estimación de esfuerzo (tamaño del software) Puntos del proyecto	(PB: Product Backlog)	80 puntos
PM	Velocidad en puntos del equipo por mes	(PM: Puntos por mes)	20 puntos
	Duración estimada del proyecto en meses	Puntos del proyecto / Velocidad en puntos del equipo por mes	4 meses
C1	Costo del equipo de trabajo y otros (en millones mensuales)	Salarios del equipo de desarrollo y otros costos como servidores, plantillas, impuestos, etc	56.600.000
CP	Costo aproximado por punto	Costo aproximado por punto = (C1) / PM. Costo aproximado por punto = 56.600.000/20	2.830.000
	Costo estimado del proyecto de software	Costo estimado = Puntos del proyecto x Costo de un punto Costo estimado = 80 x 2.830.000	226.400.000

Importante adicionar utilidad estimada. Ejemplo: 30%, 40%, 100%, 300% ...

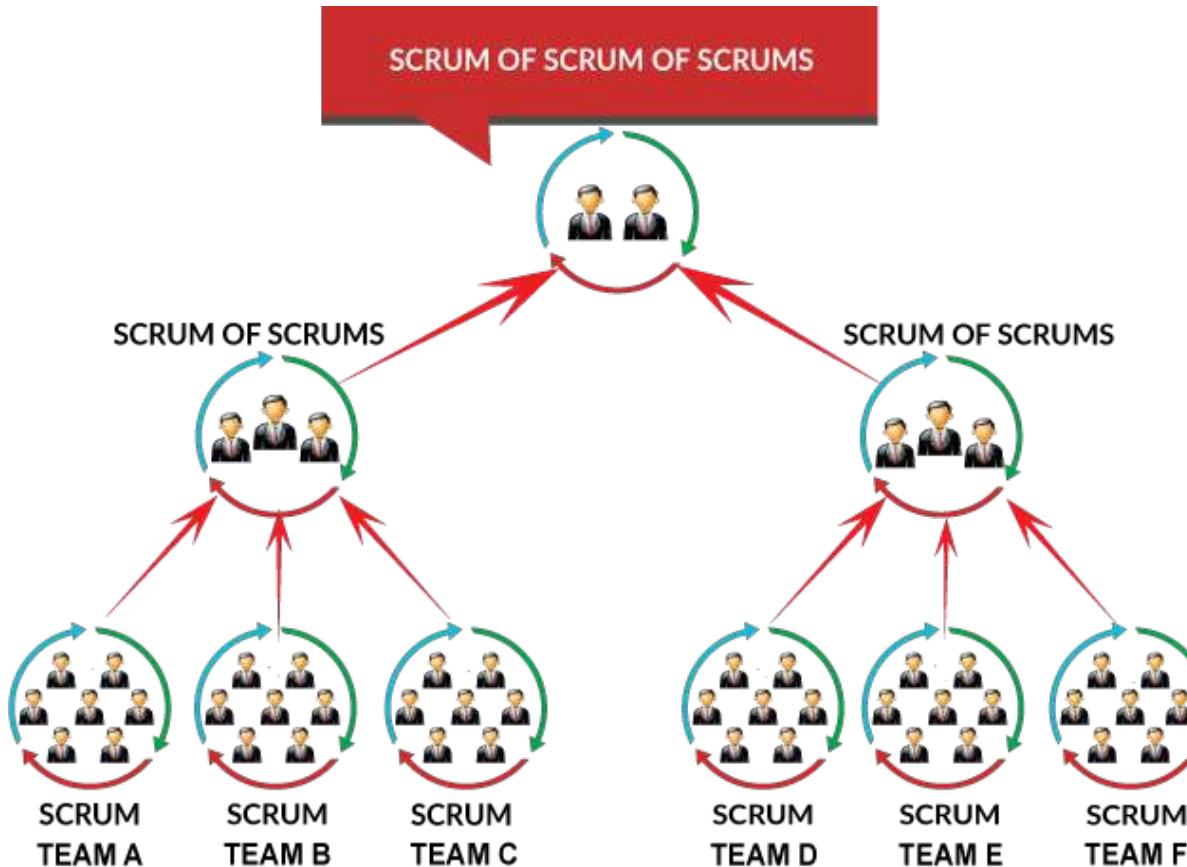
Precio = costos + utilidad estimada



Costo de una App en las metodologías ágiles

Resumen

Precio estimado de un proyecto de software							
	Valor hora	#horas mes	Valor mes		Valor	Meses	Total
Desarrollador 1	30.000,00	160	4.800.000,0	Costo mensuales	56.600.000,0	1	56.600.000,00
Desarrollador 2	30.000,00	80	2.400.000,0				
Desarrollador 3	30.000,00	160	4.800.000,0	Subtotal			56.600.000,00
Product Owner	30.000,00	160	4.800.000,0	Puntos por mes			20
Scrum Master	40.000,00	160	6.400.000,0	Total puntos			80
Tester	30.000,00	160	4.800.000,0	Costo estimado de un punto			2.830.000,00
Infraestructura	30.000,00	80	2.400.000,0	Costo estimado del proyecto			226.400.000,00
Diseñador	30.000,00	80	2.400.000,0	Utilidad (40%)			90.560.000,00
	Subtotal		32.800.000,0	Precio estimado del proyecto (sin impuestos)			316.960.000,00
				IVA 19%			60.222.400,00
Asesor 1	40.000,00	40	1.600.000,0	Otros impuestos (7%)			22.187.200,00
				Precio estimado del proyecto			399.369.600,00
Gerente	40.000,00	160	6.400.000,0				
Secretaria	15.000,00	160	2.400.000,0				
Auxiliar	15.000,00	160	2.400.000,0				
Servicios públicos, arriendo			9.000.000,0				
Hosting, librerías, templates, etc			1.000.000,0				
Otros costos fijos			1.000.000,0				
Costos mensuales			56.600.000,0				



Preguntas

1. ¿Qué ha hecho mi equipo hasta ahora desde la última vez que nos reunimos que podría afectar a otros equipos?
2. ¿Qué hará mi equipo antes de que nos volvamos a encontrar que pueda afectar a otros equipos?
3. ¿A qué problema se enfrenta mi equipo que podría obtener la ayuda de otro equipo para resolverlo?



Prácticas ágiles

Resumen

En las empresas es común que los desarrolladores trabajen en más de un proyecto. ¿Cómo se maneja esto en Scrum?

Disponibilidad de los desarrolladores (Sprint de dos semanas 80 horas)

	Desarrollador 1	Desarrollador 2	Desarrollador 3
Ceremonias Scrum	12	12	12
Perdidas por productividad	10	10	10
Otros proyectos	0	16	32
Subtotal	22	38	54
Horas disponibles	(80-22) 58	42	26

Puntos planeados en el sprint 14P 11P 8P

Si un punto equivale a 4 horas 56 44 32





Preguntas ?

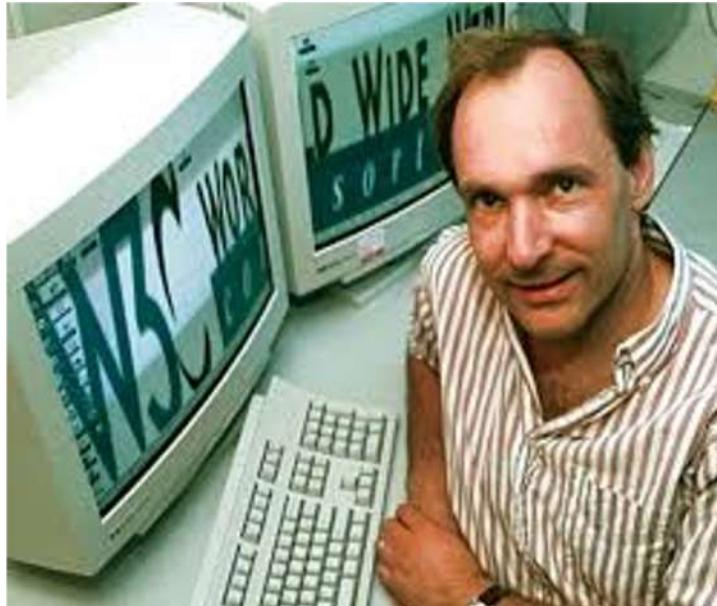


Aplicaciones WEB

El World Wide Web (WWW)

Resumen

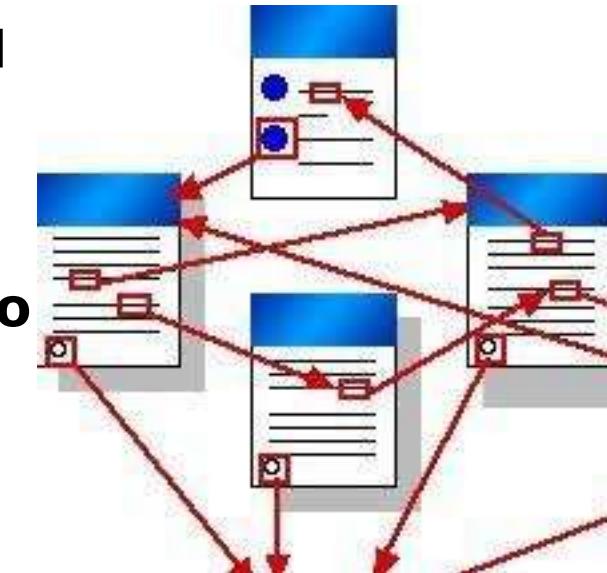
¿Cómo inició? ¿Por qué se creó?



Muchos programas + muchos formatos + muchos protocolos = **caos**

Iniciativa World Wide Web (WWW) (web)
Tim Bernes-Lee 1.989 y asociados del CERN
Nuevo sistema de información

Propósito orden
Programa llamado Browser
Idea central: Metáfora de **Hipertexto**
e Hipermedia



¿Quién es el dueño del Web?

No existe un individuo o entidad sola que sea dueño del web, es un consorcio internacional de carácter abierto y libre, pero existen ciertas entidades a nivel mundial, que lideran e impulsan su organización y desarrollo.

En los Estados Unidos, existe el WWW (W3); Consortium W3 en el MIT
(<https://www.w3c.org/>)

Universidad
del Valle

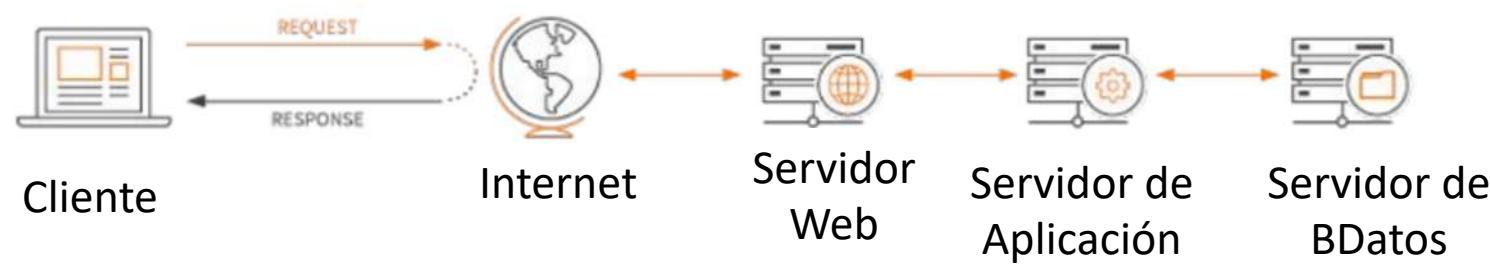


En el consorcio W3C trabajan conjuntamente para desarrollar estándares para la Web.

Arquitectura (Cliente Web - Servidor Web)

Es un extensión de la arquitectura cliente/servidor tradicional, donde las conexiones son **sin-estado** (Stateless Connections)

- Abrir una conexión
- Obtener un recurso (ejecutar un archivo y obtener la respuesta)
- Cerrar la conexión



Soporte para protocolos de transferencia de datos

HTTP://WWW.GOOGLE.COM

-  **HTTP:** (Hypertext Transfer Protocol) Protocolo nativo del web.
-  **HTTPS:** (Hypertext Transfer Protocol Secure)
-  **FTPS:** (File Transfer Protocol Secure) Protocolo de transferencia de archivos.
-  **GOPHER:** Universidad de Minnesota, Campus Informative Service Protocol.
-  **NNTP:** Protocolo usado para distribuir artículos colocados en Usenet.
-  **FILE:** Protocolo para archivos locales.
-  **TELNET:** Protocolo de comunicaciones basado en comandos de texto.

Anatomía de una URL

`https://www.univalle.edu.co:443/info/estudiantes.html`



Direcciones web: (ICANN)

(no patrocinados) (Dominios de alto nivel)

(.edu, .com, .gov, .net, .mil, .org, .int)

(.biz [www.neulevel.biz]) para negocios

(.info [www.afiliadas.info]) para todos los usuarios

(patrocinados)

(.aereo, .mobi, .coop, .jobs, .museum, .pro, .travel

Infraestructuras: .arpa, .root

En Fase de inicio: .post, .tel

Propuestos: .asia, .cym, .geo, .kid, .kids, .mail,
.sco, .web, .xxx

Borrados/retirados: .nato

Reservados: .example, .invalid, .localhost, .test

Pseudominios: .bitnet, .csnet, .local, .onion, .uucp, ws

gTLD, nuevos a partir de 2014 <http://newgtlds.icann.org/en/program-status/delegated-strings>

.futbol .click .help .cards. lltda .vision .productions .berlin etc.

Paises (<http://www.iana.org/root-whois/index.html>)

Casos .co , .tv, io, ai

Peticiones HTTP o HTTPS

Una petición de envío o solicitud de información entre el navegador y el servidor web se hace usando GET o POST

PETICION

- Métodos de petición ([GET](#), [POST](#))
- Uniform Resource Identifier (URI) específicamente una ([URL](#))
- Versión de HTTP que entiende el cliente
- Headers (información adicional respecto a la transferencia)



RESPUESTA

- Versión del protocolo y el código de estado
- Respuesta a los headers
- Respuesta al body

Tarea: Investigar la diferencia entre GET y POST

Aplicaciones Web

90-95 “Sitios Web”

- Páginas que tenían texto y gráficos con enlaces

Hoy en día..... “Aplicaciones Web” WebApps

- Capacidad de realizar lógica de la aplicación.
- Funciones para interacción con el usuario final
- Integradas con bases de datos
- Integradas con aplicaciones de negocios
- Múltiples tecnologías trabajando juntas
- Énfasis en las interfaces de usuario
- Arte y tecnología
- Tipos de App web : Estáticas, dinámicas, SPA, MPA, Animadas, CMS, Ecommerce y Portales
- Nuevas características de las Apps PWA, SaaS, Smart APPs, ...

Aplicaciones Web

Características Frecuentes

- Intensidad de red
- Carga impredecible
- Concurrencia
- Alto desempeño
- Disponibilidad (24/7/365)
- Gobernada por los datos
- Evolución continua (tecnologías, dispositivos, etc)
- Inmediatez
- Seguridad
- Facilidad de interacción con el usuario





Preguntas ?





Ítems que deberían tener en cuenta para la primera presentación de avance del proyecto

Fecha de entrega mayo 24 de 2023

1. Título del Sistema
2. Objetivos (principales funcionalidades solicitadas)
3. Empresas a las que le sirve el sistema
4. Idea del proyecto: Presenta un resumen de no más de 100 palabras que resumen de la idea del proyecto.
5. Fechas críticas: Define las fechas críticas de la aplicación tales como inicio y finalización del proyecto y fechas estimadas de los releases. (Entrega final del proyecto junio 28)
6. Miembros del equipo de desarrollo: El equipo se organiza a través de asignación de los 3 roles básicos de la metodología distribuidos entre los integrantes del grupo
7. Product backlog (o enlace a donde este): Backlog priorizado y estimado.
8. Prácticas ágiles a usar durante el desarrollo: Presenta un listado de las prácticas ágiles que acordaron usar durante el desarrollo.
9. Tecnologías a usar : Define mediante un listado todas las tecnologías de desarrollo de la aplicación
10. Diagrama de bloques(módulos de la aplicación): Presenta un diseño inicial de los módulos de la aplicación que representan el estilo arquitectural de la aplicación.
11. Modelo de datos (modelo relacional): Presenta un modelo de datos inicial.
12. Diseño de las interfaces de la aplicación (Interfaces del primer sprint)
13. Release plan: Presenta un release plan inicial donde presenta los sprints a realizar, HU de cada sprint, responsable de cada HU y fecha estimada de entrega de los sprints y releases del proyecto.
14. Detalles de la Historias de usuario : Presenta el detalle de las HU de al menos el primer sprint en un formato definido.
15. Acrónimos: Presenta una lista de los acrónimos usados con su respectiva definición (si los hay)
16. Primer Sprint del proyecto.



Fecha de Exposiciones, Taller y Proyecto

Grupo	Tema	Fecha exposición
	Pruebas unitarias en Python y Django	Mayo 3 de 2023
	Estándares de codificación en Python y Django	Mayo 3 de 2023
	Taller Django	Mayo 10 de 2023
	Tablero Kanban	Mayo 17 de 2023
	Chrome DevTools	Mayo 17 de 2023
	Primera entrega del proyecto	Mayo 24 de 2023
	Git y git flows (incluir git tags)	Mayo 31 de 2023
	*Web services con Django	Mayo 31 de 2023
	Motion UI	Junio 7 de 2023
	*WAF (Web Applications Firewall)	Junio 7 de 2023
	Desarrollo de aplicaciones web usando ChatGPT	Junio 14 de 2023
	Entrega final del proyecto	Junio 28 de 2023



Preguntas ?





Modelo de Despliegue

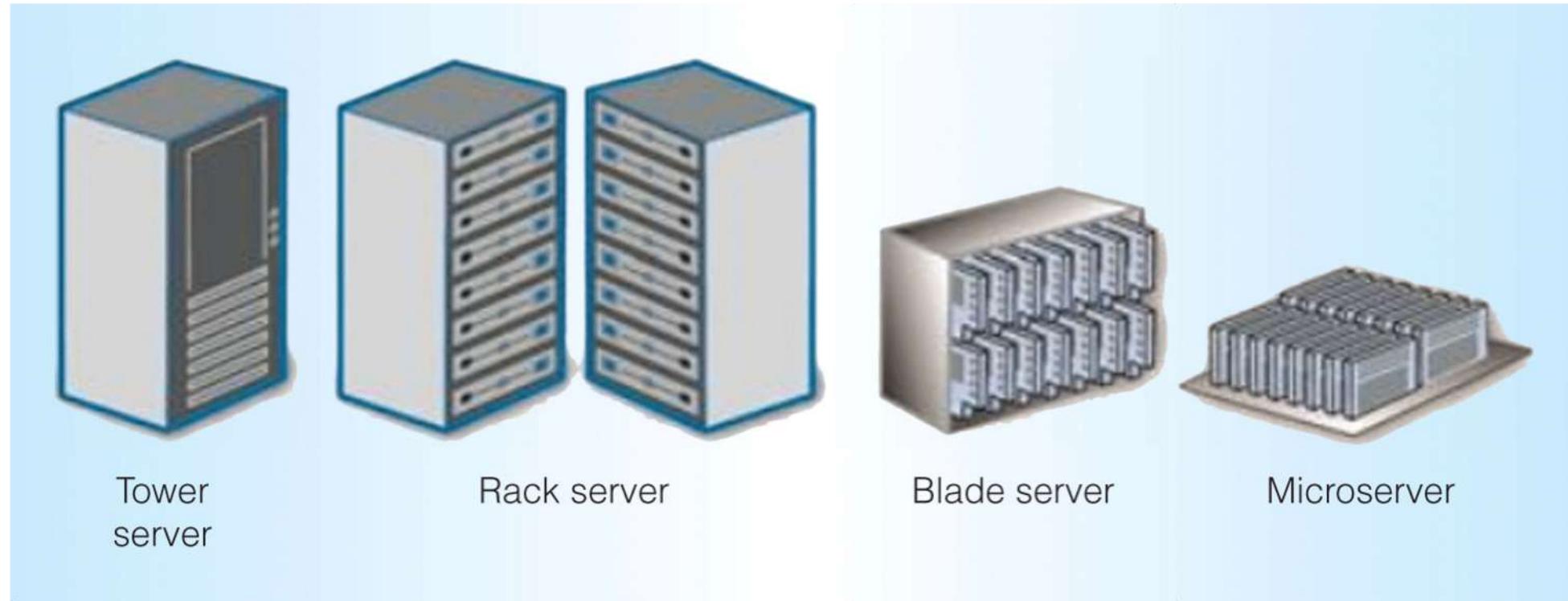
Representa la plataforma o el conjunto de servidores requeridos por una aplicación de software para ponerla en funcionamiento para ser usada por los usuarios finales.



Modelo de despliegue

Servidor (Host)

Evolución del hardware para servidores



Un equipo

Data Center local (on-premise)

Data Center externo

Cloud Data Center

Donde colocar los servidores ?

Modelo de despliegue



Data Center



Centro de control de Red (NOC)

Desarrollo de Software I

Modelo de despliegue

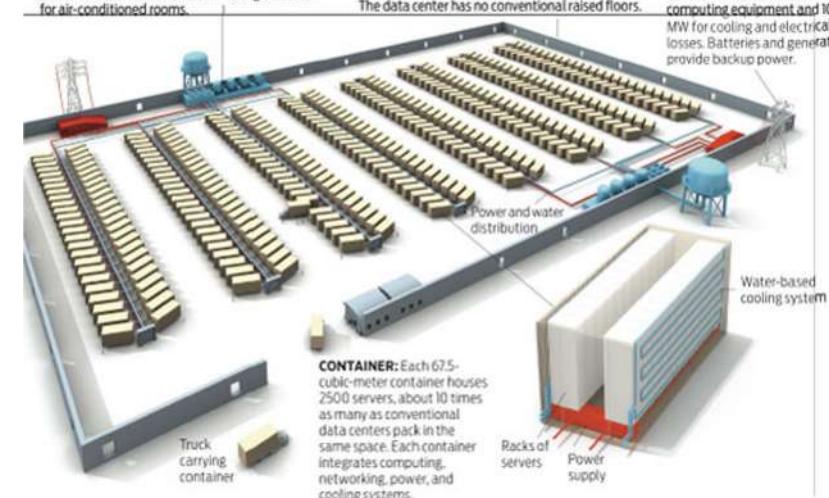
Centro de computación en Nube



COOLING: High-efficiency water-based cooling systems—less energy-intensive than traditional chillers—circulate cold water through the containers to remove heat, eliminating the need for air-conditioned rooms.

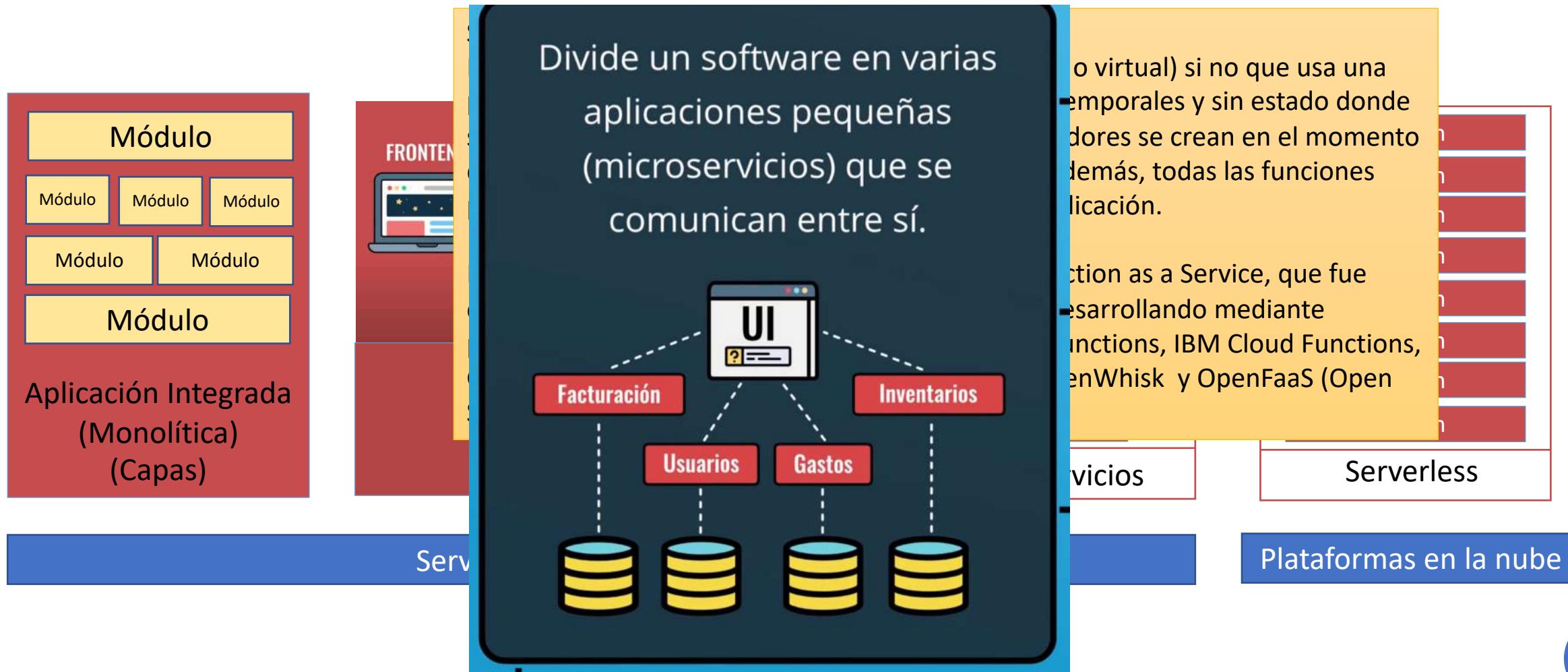
STRUCTURE: A 24 000-square-meter facility houses 400 containers. Delivered by trucks, the containers attach to a spine infrastructure that feeds network connectivity, power, and water. The data center has no conventional raised floors.

POWER: Two power substations feed a total of 300 megawatts to the data center, with 200 MW used for computing equipment and 100 MW for cooling and electrical losses. Batteries and generators provide backup power.



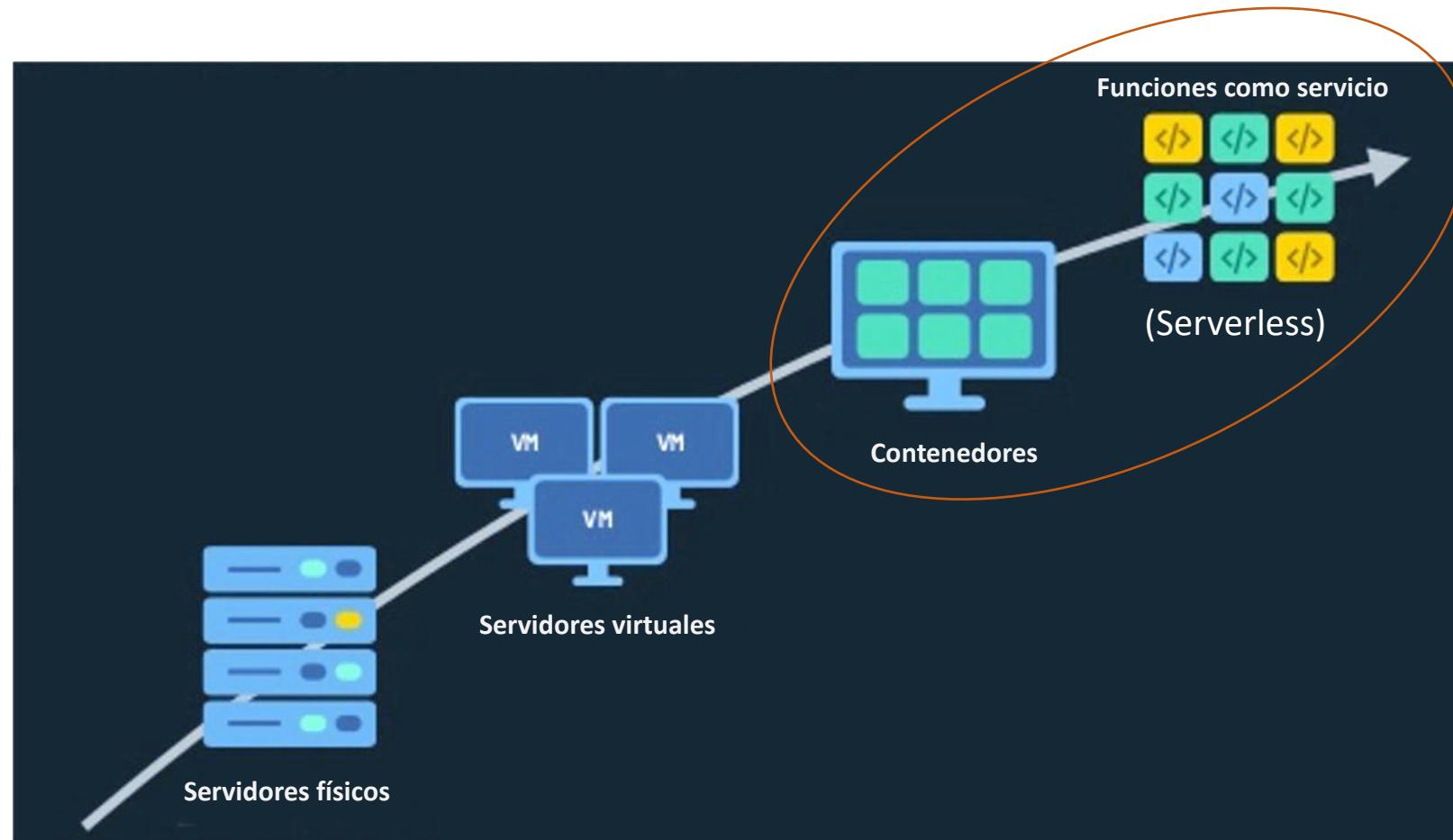
Modelo de despliegue

¿Como construir aplicaciones para ser desplegadas: Estilo arquitectural



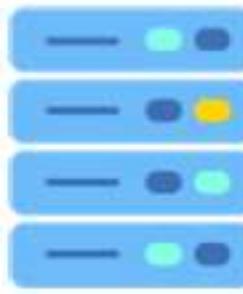
Modelo de despliegue

Evolución de los servidores para despliegue de aplicaciones

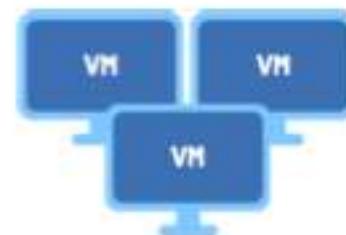


Modelo de despliegue

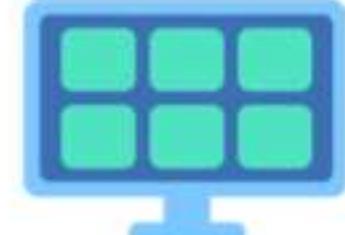
Evolución de los servidores para despliegue de aplicaciones



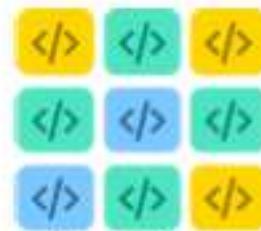
Servidor físico



Máquina virtual



Contenedores



Serverless

Code

App Container

Language Runtime

Operating System

Hardware

Desarrollo de Software I

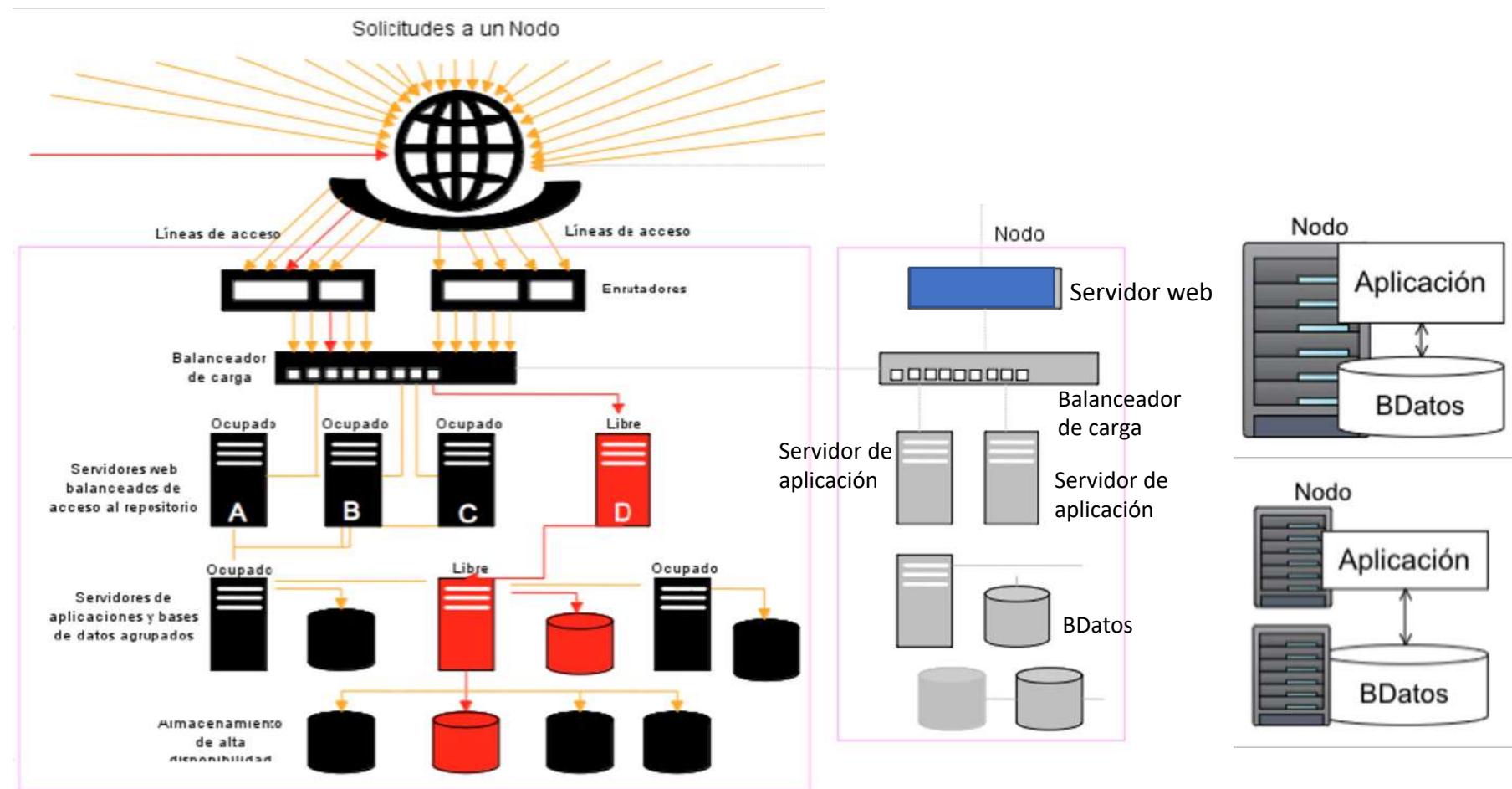
Modelo de despliegue

Ambientes para desarrollo, pruebas y despliegue de aplicaciones



Modelo de despliegue

Nodos para despliegue de aplicaciones



Modelo de despliegue

Como calcular el tamaño de los servidores de los nodos para despliegue de aplicaciones (RAM, Procesadores, Espacio de almacenamiento, Ancho de banda)

Factores a tener en cuenta para determinar el tamaño de un servidor:

- Carga (tráfico) : Número de peticiones que depende del número de usuarios concurrentes
- Recurso que requiere la aplicación: RAM, Procesadores, Almacenamiento, Ancho de banda
- Tipo de Aplicación: Cantidad de recursos que consume. Nota: si la aplicación usa muchos videos estos se deben colocar en un servidor de videos como youtube, Vimeo, etc.
- Escalabilidad requerida : Crecimiento por demanda
- Presupuesto

Modelo de despliegue

Como calcular el tamaño de los servidores de los nodos para despliegue de aplicaciones
(RAM, Procesadores, Espacio de almacenamiento, Ancho de banda)

Ejemplo: Características típicas en VM

1 Gigas RAM 1 o 2 Procesadores	2 Gigas RAM 2 Procesadores	4 Gigas RAM 2 Procesadores	8 Gigas RAM 2 Procesadores	16 Gigas RAM 2 Procesadores
4 Gigas RAM 4 Procesadores	8 Gigas RAM 4 Procesadores	16 Gigas RAM 4 Procesadores	32 Gigas RAM 4 Procesadores	64 Gigas RAM 4 Procesadores
8 Gigas RAM 8 Procesadores	16 Gigas RAM 8 Procesadores	32 Gigas RAM 8 Procesadores	64 Gigas RAM 16 Procesadores	128 Gigas RAM 16 Procesadores

Otros aspectos que impactan :

- Número de aplicaciones en el mismo servidor
- Velocidad del procesador
- Tipos de discos (SSD)

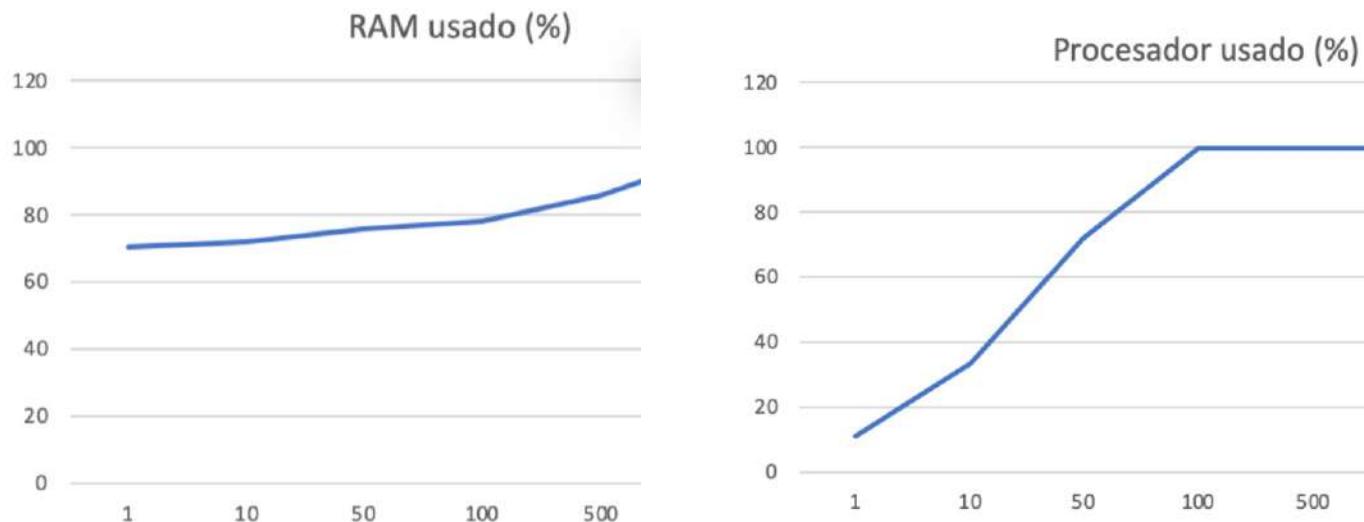
Las pruebas carga de las aplicaciones ayudan a determinar las características de los servidores requeridos

Desarrollo de Software I

¿Cómo calcular las características de un servidor(es) para una Aplicación?

Hay herramientas te permiten simular el tráfico de usuarios en una aplicación web y medir cómo responde el servidor.

Ejemplo: Equipo base: 4 CPUs y 16 de RAM



Proyección de las caraterísticas de los servidores

Usuarios	CPUs	RAM en GB
50	4	16
100	8	32
500	12	64
1000	16	72

Las pruebas de las aplicaciones determinan las características de los servidores requeridos.

Herramientas para medir recursos (monitorear consumos o uso de recursos en los servidores)

Apache JMeter, LoadRunner de Micro Focus, Gatling, Solarwinds, etc

<https://www.solarwinds.com/server-application-monitor/use-cases/server-capacity>



Modelo de despliegue

Despliegue de aplicaciones (**Deploy**)

El despliegue de aplicaciones consiste en copiar los archivos de una aplicación en un servidor o nodo donde funcionará la aplicación. (configurado)

Que archivos copiar:

- Programas ejecutables o scripts
- Librerías requeridas con sus versiones específicas
- Otras librerías o paquetes (Ej: librerías gráficas, etc)
- Archivos de datos requeridos
- Bases de datos y drivers de conexión

Modelo de despliegue

Despliegue continuo: Despliegue por Release, por Sprint o Hot fix

Métodos para hacer Deploy:
Manual

Automático: **IaC** (Infrastructure as a Code)

Instalar desde un repositorio (EJ. Git pull)
`git clone https://github.com/github/miapp.git`



Infraestructure as a Code (IaC)
Infraestructura como código (IaC) significa administrar su infraestructura de TI mediante archivos de configuración.



Los procesos de desarrollo ágil intentan que el “Deploy”
cada vez sea más automático

Despliegue de aplicaciones Web: Estrategias para actualizar el servidor

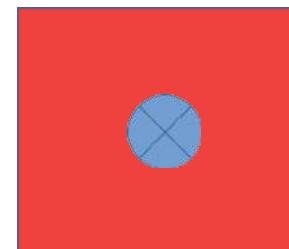
Tipo Stand Alone (Stop-Start)

- Se basa en tener un servidor (de producción) sobre el cual se realizan los cambios, **parar el servidor, realizar los cambios y luego reiniciar el servidor.**
- Cuando es por primera vez, puede ser un proceso adecuado.
- No se recomienda esta estrategia cuando el servidor está en producción.
- Al parar el servidor todos los usuarios no tendrán acceso al servicio.

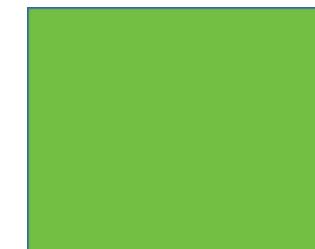
En funcionamiento



No disponible

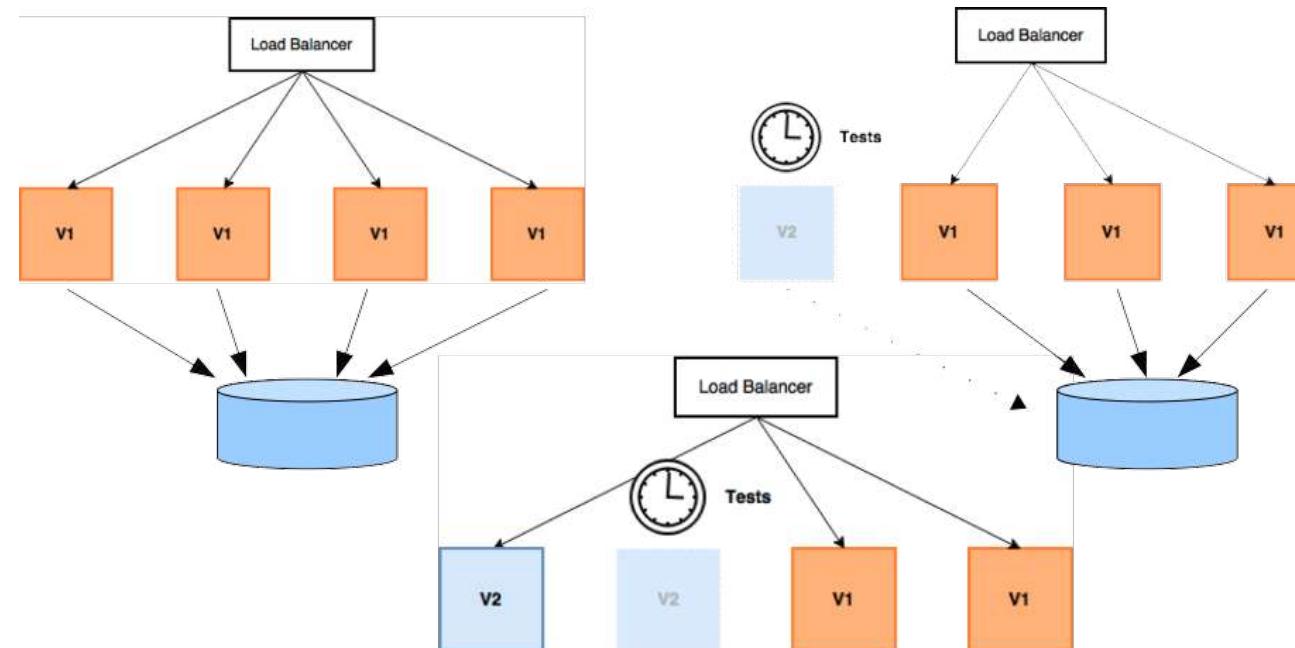


En funcionamiento



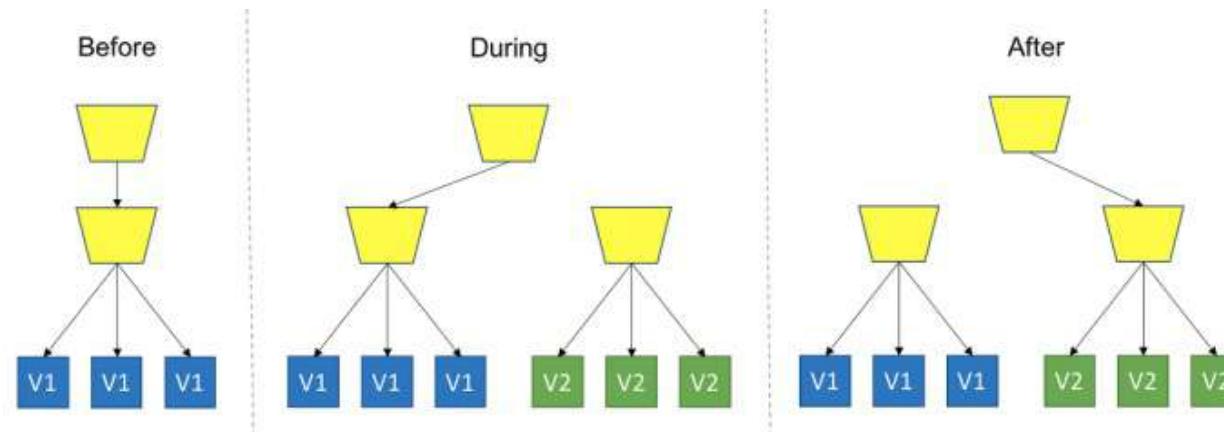
Despliegue de aplicaciones Web: Estrategias para actualizar el servidor

Rolling upgrade: Se basa en tener una capa de balanceo de la aplicación. Se desconecta un servidor del balanceador, se actualizase y se prueba, si funciona bien, se conecta al balanceador de nuevo y se repite el proceso hasta actualizar todos los servidores.



Despliegue de aplicaciones Web: Estrategias para actualizar el servidor

Despliegue Blue/Green: Duplicar temporalmente la infraestructura y desplegar todo a la vez

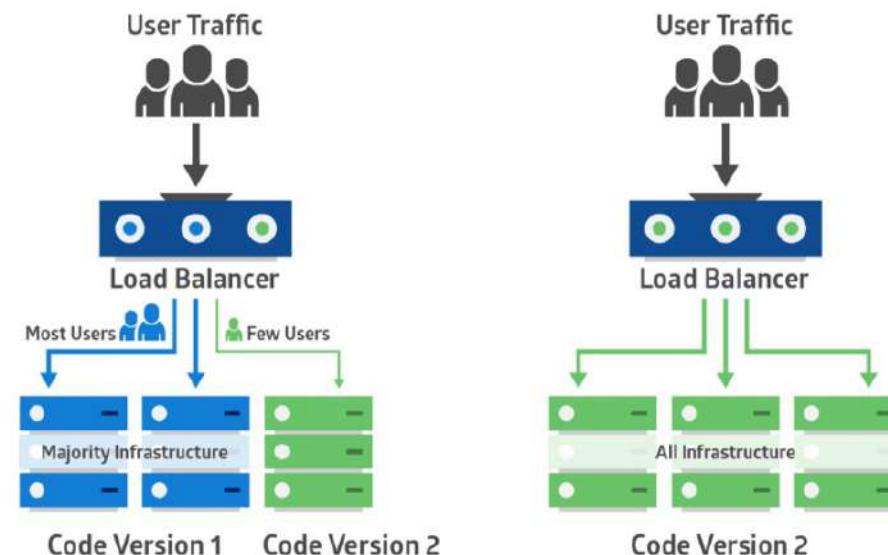


Esta estrategia también se puede aplicar cuando es un solo servidor.
Usar contenedores es muy práctico para este tipo de despliegue.
También le dan nombres como Despliegue A/B o Red-Black.

Despliegue de aplicaciones Web: Estrategias para actualizar el servidor

Canary Deployment: "rolling out releases" Despliegue del canario es un patrón de despliegue que se aplica a un subconjunto de usuarios o servidores.

- La idea principal es primero desplegar los cambios en un conjunto pequeño de servidores, probar con algunos usuarios y luego continuar el despliegue hacia el resto de usuarios y servidores.
- El despliegue "canary" sirve como sistema indicador de alerta temprana, si el despliegue del "canario" falla, el resto de servidores no se ven impactados.





Preguntas ?

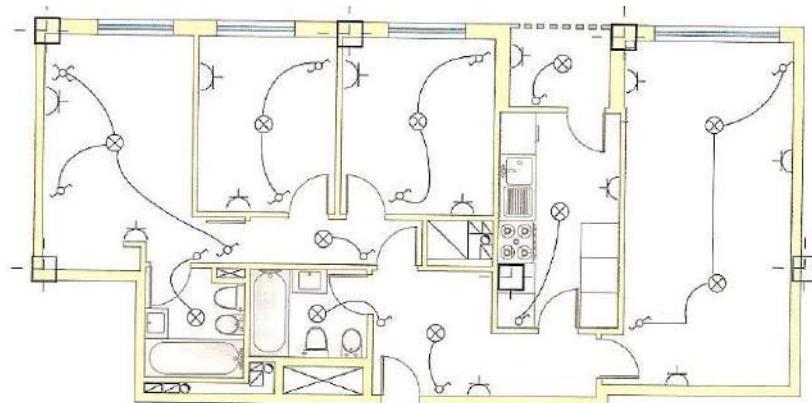
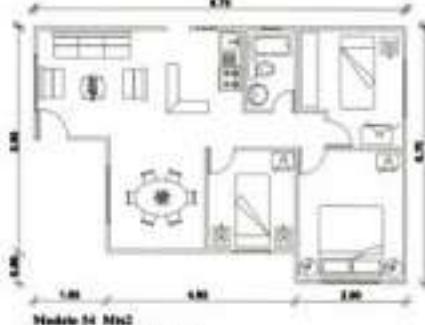




Introducción a las Arquitecturas Ágiles de Software

Desarrollo de Software I

Arquitectura ágil de software



- 48
- ⊕ punto de luz en el techo
 - ⊖ interruptor de cruce
 - ⊖ interruptor sencillo
 - ⊖ toma de corriente de usos varios
 - ⊖ interruptor conmutado
 - ⊖ toma de corriente de lavadora y lavavajillas

Arquitectura ágil de software



1



2

Adopción de las Arquitecturas

El desarrollo de software ágil ha sido adoptado por una gran cantidad de empresas en un esfuerzo por reducir costos e incrementar su habilidad para manejar los cambios en un mercado de condiciones cambiantes.

Arquitectura como resultado de usar las prácticas ágiles

- Las Historias de Usuario y las restricciones (requerimientos no funcionales) son base de las arquitecturas ágiles
- La arquitectura emerge sprint tras sprint como resultado de constantes y pequeñas mejoras.
- Una arquitectura ágil es un proceso iterativo que tiene un enfoque modular en la creación de un sistema (su arquitectura)

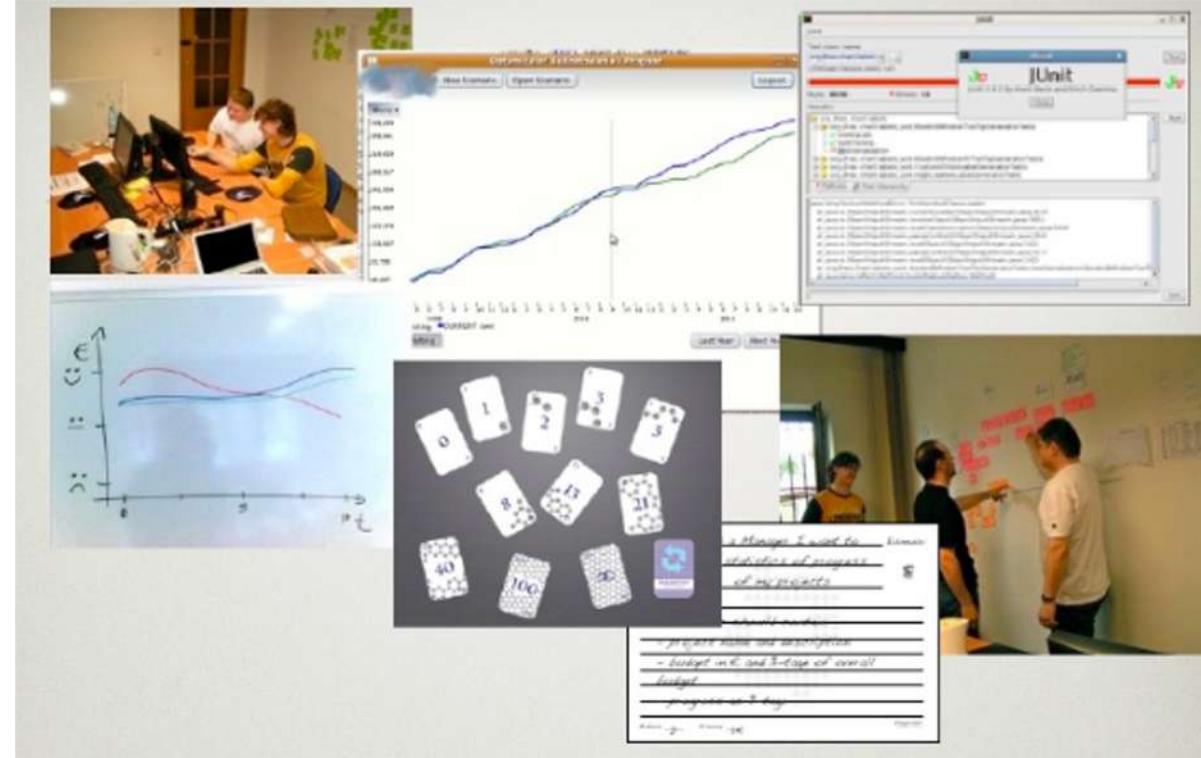


La arquitectura de software ágil es el resultado de la aplicación de las prácticas ágiles.

Arquitectura ágil de software



Arquitectura ágil de software

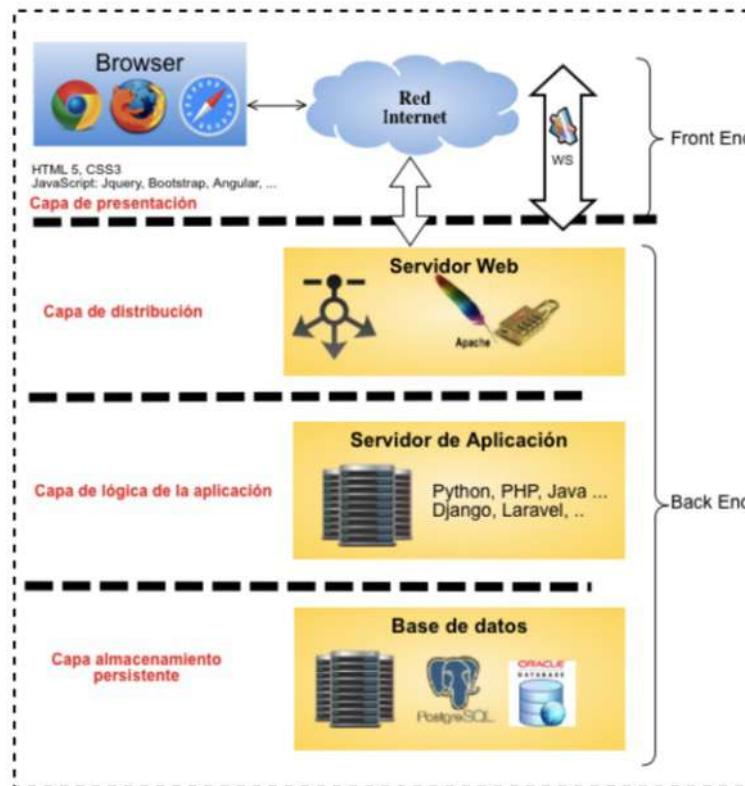


La Arquitectura ágil es una actividad en equipo que se logra iteración tras iteración y que integra estrategia, negocios y tecnología.

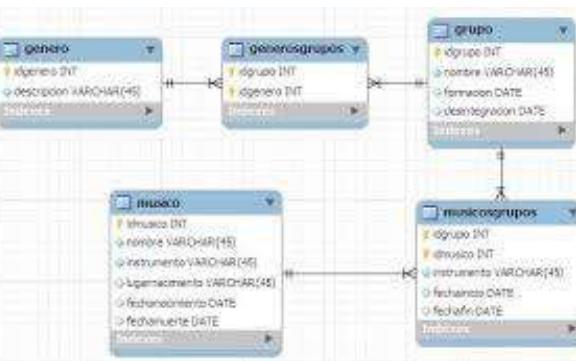
Iniciando un proyecto de software usando prácticas ágiles

Modelos arquitecturales del sistema

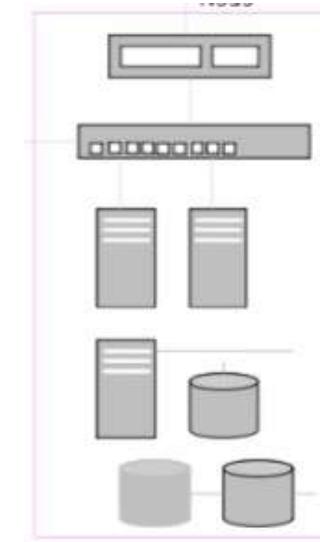
“Simple Sketches for Diagramming your Software Architecture “



Estilo arquitectural de la aplicación



Modelo de datos (inicial)



Modelo de despliegue

Valores en el diseño de Arquitecturas ágiles

- Simplicidad en el diseño
- Sistemas que funcionen
- Mejoramiento continuo
- Trabajo en equipo
- Comunicación
- Confianza
- Satisfacer las necesidades de los stakeholders
- Aprendizaje continuo
- Diseñe cosas para probar
- Modular o desacoplado
- Evitar redundancias
- Diseñada pensando en los atributos de calidad: Testeabilidad, Mantenibilidad, Despliegue, Escalabilidad, Rendimiento, Seguridad y Usabilidad.

Arquitectura ágil de software

Documentación de una arquitectura ágil

Documento entre 4 y 6 páginas con enlaces

Sección	Descripción	Contenido
Descripción del producto	Descripción del producto (Máximo 100 palabras)	<ol style="list-style-type: none">1. Que hace mi producto2. Objetivos del producto y valor del negocio.3. Problemas que el producto soluciona.4. Tipos de empresas objetivo y usuarios objetivo5. [Enlaces a documentación relevante del producto o productos similares.]
Visión general de la arquitectura del producto	Descripción general de la arquitectura del sistema con sus versiones y vistas conceptuales	<p>Vistas :</p> <ul style="list-style-type: none">• Descripción general de la arquitectura (100 palabras)• Diseño de la interfaz de usuario (las más importantes)• Diseño para el desarrollo (equipos y herramientas usadas para el desarrollo, control y seguimiento)• Principales Componentes del sistema (Diagrama de Bloques o Diagrama de Microservicios)• Modelo de datos (alto nivel)• Diagrama de despliegue

Documentación de una arquitectura ágil

Documento entre 4 y 6 páginas

Sección	Descripción	Contenido
Requerimientos (Historias de usuario)	Descripción de alto nivel de las funcionalidades	<ul style="list-style-type: none">• Product Backlog• Detalles de las historias de usuario (link)
Prácticas ágiles en el proceso de desarrollo	Descripción corta de las prácticas ágiles usadas	<ul style="list-style-type: none">• Resultados de las prácticas ágiles realizadas (link) <p>Para las ceremonias videos o actas de lo presentado o realizado en la reunión</p>
Acrónimos	Acrónimos usados en el documento	<ul style="list-style-type: none">• Descripción corta



Preguntas ?



Equipos para el proyecto del curso y exposiciones:

Trabajo en equipos del proyecto del curso



Desarrollo I

Economy of the
European Union

Gracias

