



## UNIVERSIDAD TECNOLÓGICA DE LA SELVA

**División:**

Tecnologías de la información

**Carrera:**

Desarrollo de Software Multiplataforma

**Materia:**

Desarrollo de Aplicaciones Web

**Unidad Temática:**

Unidad I. Fundamentos de arquitecturas de software

**Actividad de evaluación:**

Investigación

Nombre del alumno:	Matrícula:
Leonardo Antonio Guillén Navarro	091910039

**Grado:** 7°

**Turno:** Vespertino

**Grupo:** A

**Nombre del profesor:**

MC. Armando Méndez Morales

**Lugar y Fecha de entrega:**

Ocosingo, Chiapas, 8 de septiembre de 2021

## Arquitectura de Software

La arquitectura a nivel de software es el diseño de más alto nivel de la estructura de un sistema.

Los objetivos son aquellos perfilados para el sistema de información, pero no solamente los de tipo funcional.

también otros objetivos como el mantenimiento, la auditoría, flexibilidad e interacción con otros sistemas de información. Por ejemplo, no es aceptable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real.

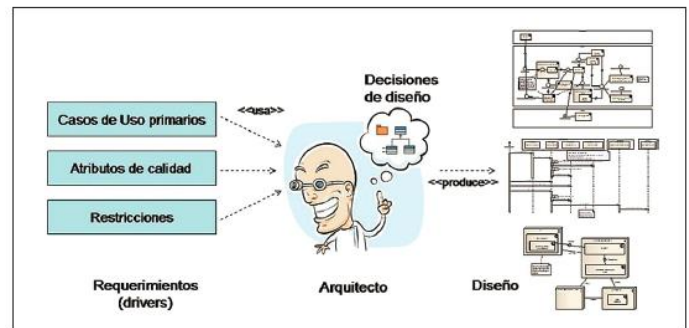
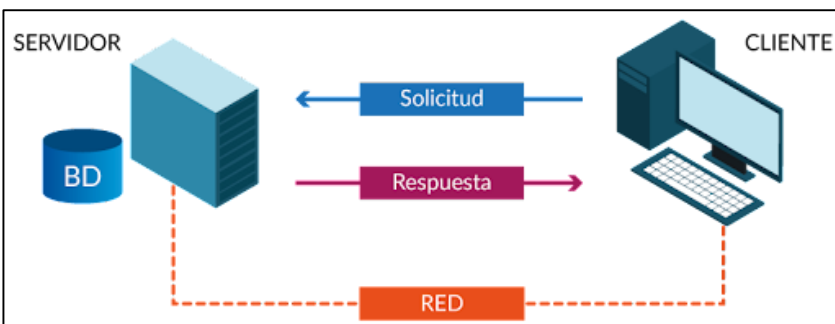
La arquitectura de software define de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y comunicación entre ellos.

### Síntesis.

La arquitectura de software se refiere al grado más alto de la estructura de un sistema, donde son consideradas elementos que harán de este sistema mayor profesionalidad como lo es el mantenimiento, auditoría, flexibilidad e interacción con otros sistemas.

### Ejemplos.

- Descomposición Modular: Donde el software se estructura en grupos funcionales muy acoplados.
- Cliente-servidor: Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.



## Architectural Pattern

Un patrón arquitectónico es un concepto que resuelve y define algunos elementos cohesivos esenciales de una arquitectura de software. Aunque un patrón arquitectónico transmite una imagen de un sistema, no es una arquitectura. Es una solución general y reutilizable para un problema común en la arquitectura de software dentro de un contexto dado.

Los patrones arquitectónicos abordan varios problemas en la ingeniería de software. Como las limitaciones de rendimiento del hardware de la computadora, la alta disponibilidad y la minimización de un riesgo comercial.

## Design Pattern

Son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción de interfaces. Resulta ser una solución a un problema de diseño. Para que una solución sea considerada patrón de diseño debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

## Síntesis

un patrón arquitectónico es una solución a un problema común dentro de la arquitectura de software.

Un patrón de diseño son técnicas estandarizadas en la cual implica conceptos para realizar abstracciones poder



agrupar código, componentes, tal que puedan ser reutilizables y mantenibles a lo largo del tiempo

Ejemplos de Patrones Arquitectónicos.

- Patrón de capas
- Patrón de intermediario
- Patrón cliente servidor
- Patrón de igual a igual
- Patrón maestro esclavo
- Patrón de bus de eventos
- Patrón de filtro de tuberías
- Patrón vista controlador

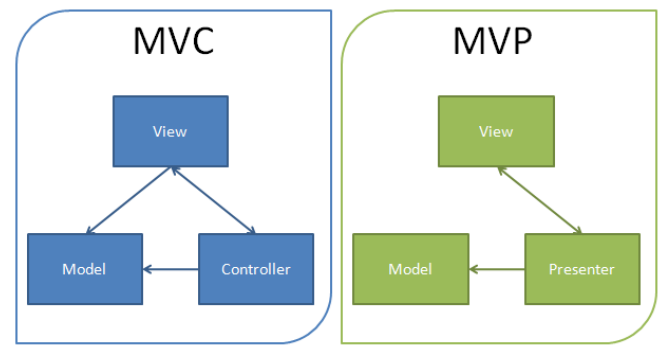
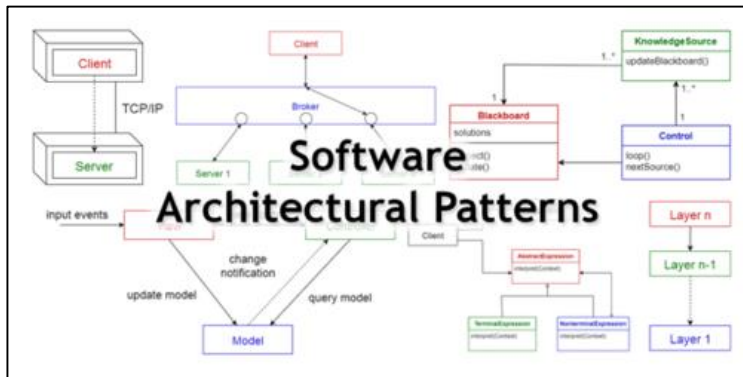
Ejemplos Patrones de diseño.

- Factory Method
- Adapter
- Singleton
- Builder
- Observer
- State
- Strategy

Diferencias.

Los patrones arquitectónicos son similares a los patrones de diseño de software pero tienen un alcance más amplio.

- Patrón de Arquitectura: expresan un esquema organizativo estructural fundamental para sistemas de software.
- Patrón de diseño: expresan esquemas para definir estructuras de diseño (o relaciones) con las que construir sistemas de software.



## Patrones de Diseño

- Clasificación

Ámbito	Propósito		
	Patrones de Creación	Patrones estructurales	Patrones de Comportamiento
Clase	<ul style="list-style-type: none"> <li>Factory Method</li> </ul>	<ul style="list-style-type: none"> <li>Adapter (de clases)</li> </ul>	<ul style="list-style-type: none"> <li>Interpreter</li> <li>Template Method</li> </ul>
Objeto	<ul style="list-style-type: none"> <li>Abstract Factory</li> <li>Builder</li> <li>Prototype</li> <li>Singleton</li> </ul>	<ul style="list-style-type: none"> <li>Adapter (de objetos)</li> <li>Bridge</li> <li>Composite</li> <li>Decorator</li> <li>Facade</li> <li>Flyweight</li> <li>Proxy</li> </ul>	<ul style="list-style-type: none"> <li>Chain of Responsibility</li> <li>Command</li> <li>Iterator</li> <li>Mediator</li> <li>Memento</li> <li>Observer</li> <li>State</li> <li>Strategy</li> <li>Visitor</li> </ul>



## Suite de protocolos de comunicación Principales de red.

Es un grupo de protocolos que trabajan en forma conjunta para proporcionar servicios legales o integrales de comunicación de red.

La suite de protocolos pueden estar especificadas por una organización de estandarización o pueden ser desarrolladas por un proveedor.

### Síntesis.

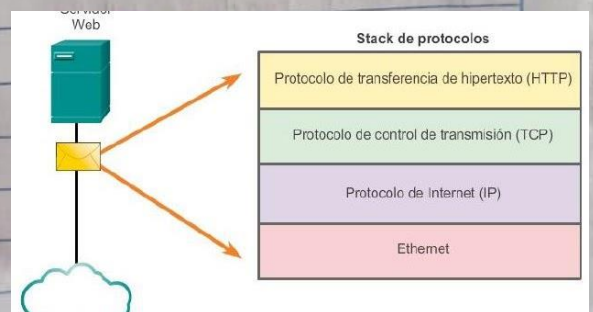
La suite de protocolos son un grupo que se especializa en trabajar de forma conjunta para poder proporcionar servicios de comunicación.

### Ejemplos.

Los protocolos IP, HTTP y DHCP son todos parte de la suite de protocolos de internet conocida como protocolo de control de transmisión.

Suites de protocolos y estándares del sector

Nombre de la capa	TCP/IP	ISO	AppleTalk	Novell Netware
Aplicación	HTTP DNS DHCP FTP	ACSE ROSE TRSE SESE	AFP	NDS
Transporte	TCP UDP	TP0 TP1 TP2 TP3 TP4	ATP AEP NBP RTMP	SPX
Internet	IPv4 IPv6 ICMPv4 ICMPv6	CONP/CMNS CLNP/CLNS	AARP	IPX
Acceso a la red	Ethernet PPP Frame Relay ATM WLAN			





## Stakeholder

Un stakeholder es el público de interés para una empresa que permite su completo funcionamiento, con público, refiriéndose a todas las personas u organizaciones que se relacionan con las actividades y decisiones de una empresa como empleados, proveedores, clientes, gobierno, entre otros.

### Síntesis.

Stakeholders son personas "interesadas" que pueden ganar algo o perder algo que ganará o perderá algo en un proyecto.

### Ejemplos

- El interés de los accionistas de una empresa está centrado en obtener beneficios derivados de su inversión, incrementando el valor inicial de esta. Mientras que el interés girará en torno a mantener su puesto de trabajo y seguir ganando dinero.

Los intereses se cruzan en la necesidad de que la empresa tenga éxito y genere beneficios.

- Los proveedores suelen ser reemplazables, pero si una administración pública no concede una licencia de actividad, no suele haber margen para actuar.



## Aplicación monolítica.

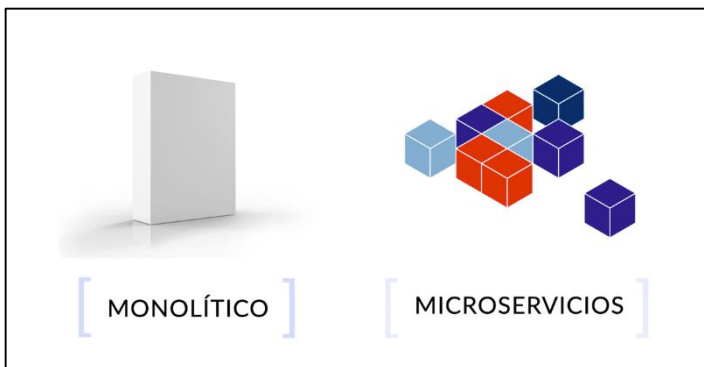
Monolito significa "trabajo" construido sobre una piedra" por lo que se utiliza para definir la arquitectura de algunos sistemas. Se refiere a como desarrollar un sistema, programa o aplicación donde todas las funcionalidades y códigos están en un solo proceso. Estas diversas funcionalidades están en el mismo código fuente y en su ejecución comparten recursos desde la misma máquina, ya sea procesamiento, memoria, bases de datos y archivos.

### Síntesis.

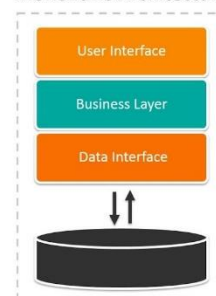
Una aplicación monolítica hace referencia a una aplicación de software en donde toda el proceso como interfaz de usuario el acceso a datos están combinados en un mismo programa.

### Ejemplos.

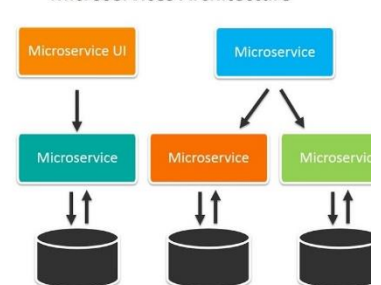
ERP = Planificación de recursos empresariales  
Son los sistemas de información gerenciales que integran y manejan muchos de los negocios.  
CRM = Gestión de clientes.  
Encargado para el desarrollo de aplicaciones parte de Microsoft llamada Dynamics 365.



### Monolithic Architecture



### Microservices Architecture





## Criterios de calidad de software.

Los criterios de calidad son aquellos que se toman en cuenta aspectos funcionales y no funcionales esenciales para garantizar la calidad en el software, además de que deben ser medibles para llevar un buen control y es necesario que sean incorporados desde el inicio del Proyecto.

### Síntesis.

Los criterios de Calidad son los que se encargan en llevar un buen seguimiento para el Proyecto, por lo cual deben ser incorporados al principio.

### Características globales para la calidad del software:

- **Funcionalidad:** Son aquellas que buscan satisfacer las necesidades del usuario.
- **Confiabilidad:** La capacidad del software de mantener su rendimiento bajo ciertas condiciones durante cierto periodo de tiempo.
- **Usabilidad:** Basada en el esfuerzo necesario para utilizar el software.
- **Eficiencia:** Basada en la relación entre el nivel de rendimiento del software.
- **Capacidad de mantenimiento:** esfuerzo necesario para realizar modificaciones específicas.
- **Portabilidad:** Basada en la calidad del software para ser transferido de un entorno a otro.



## Pruebas del rendimiento del Software.

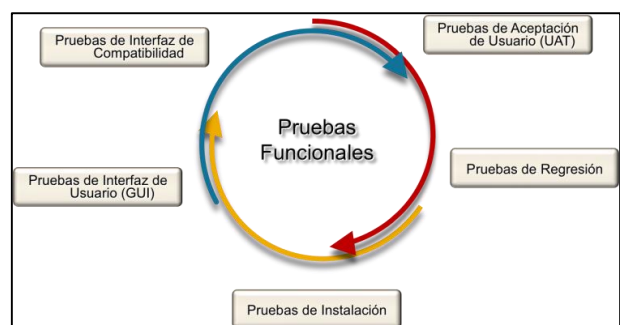
Es una técnica de prueba de software no funcional que determina cómo la estabilidad, la velocidad, la escalabilidad y la capacidad de respuesta de una aplicación se mantiene bajo una determinada carga de trabajo. Es un paso clave para asegurar la calidad del software, desafortunadamente, a menudo se ve como una reflexión posterior, en existimienta, y para comenzar una vez que se completan las pruebas funcionales, y en la mayoría de los casos después de que el código está listo para ser distribuido.

### Síntesis

Son Aquellas que se encargan de someter a un sistema a una carga de trabajo con el fin de medir su velocidad, fiabilidad y estabilidad en esas condiciones.

### Tipos de Pruebas de rendimiento.

- Prueba de carga: se realiza para observar el comportamiento de una aplicación bajo una cantidad esperada de peticiones.
- Prueba de Estrés: Se realiza para determinar la solidez de la aplicación en los momentos de carga extrema.
- Prueba de estabilidad: Se realiza para determinar si hay alguna fuga de memoria en la aplicación.
- Prueba de Pico: trata de observar el comportamiento del sistema.





## ERP

La planificación de recursos empresariales es un proceso que utilizan las empresas para gestionar e integrar las partes importantes de sus negocios. Muchas aplicaciones de software ERP son importantes para las empresas por que les ayudan a implementar la planificación de recursos al integrar todos los procesos necesarios para ejecutar sus empresas en un solo sistema.

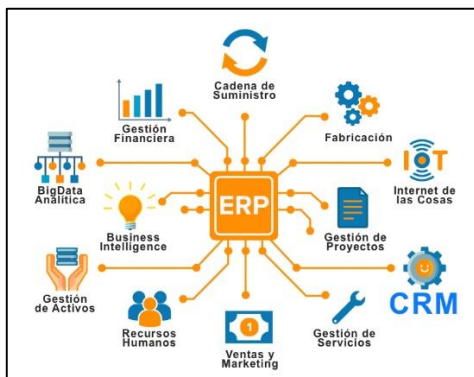
El software ERP puede integrar todos los procesos necesarios para administrar una empresa.

### Síntesis

ERP es la planificación de recursos empresariales donde se engloban las gestiones de un inventario, facturación, contabilidad, pedidos, gestión con la relación de clientes. Un software ERP integrará estas necesidades en un solo programa optimizando tiempo y dinero.

### Ejemplos de Softwares ERP.

- **Netsuite ERP:** Programa para la planificación de recursos empresariales, empresas medianas y grandes corporaciones, se incluyen gestión financiera, gestión de ingresos, facturación y gestión de inventarios.
- **TiPalti:** Es una solución ERP integral equipada con potentes módulos para la gestión de adquisiciones, proveedores, facturas y pagos. Es un punto de acceso integral para empresas para servicios financieros hasta videojuegos.





## ESB

Un bus de servicio empresarial es un patrón mediante el cual un componente de software centralizado realiza integraciones a sistemas de backend. Generalmente se implementa en un entorno de ejecución de integración especialmente diseñado y un conjunto de herramientas que garantizan la mejor productividad posible.

### Definición:

Es un modelo de arquitectura empresarial donde se gestiona la comunicación entre servicios web, el cual contiene un componente de software centralizado para el backend (traducciones de modelo de datos, conectividad profunda, direccionamiento y solicitudes).

## SOA.

La Arquitectura Orientada a Servicios de Cliente, conocida también como SOA, es un concepto de arquitectura de software que define la utilización de servicios (programas o rutinas que realizan una función específica) para dar soporte a los requisitos del negocio, permitiendo la creación de sistemas de información ampliables, versátiles y flexibles.

### Definición:

SOA es una arquitectura que brinda servicios a las empresas lo que permite la creación de sistemas de información escalables que refleja el negocio de la organización.

## Diferencia entre ESB y SOA

SOA es un modelo Arquitectónico para implementar aplicaciones basadas en servicios de acoplamiento flexible.

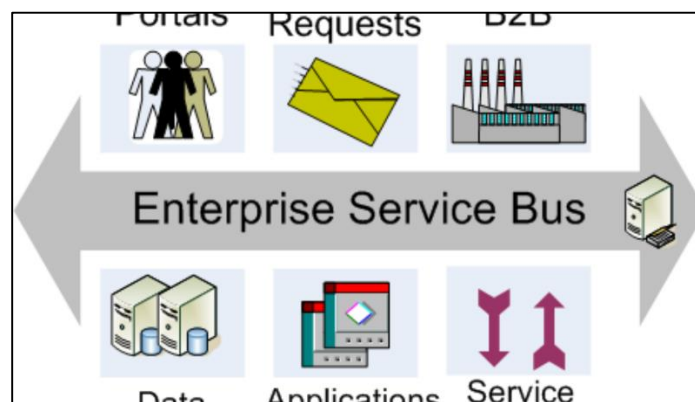
ESB es una pieza de software de infraestructura que ayuda a los desarrolladores a desarrollar servicios y comunicarse entre servicios a través de API adecuadas.

### Ejemplos de Arquitecturas SOA.

- Web services: Servicios web proporcionan una interfaz de acceso a un servicio escondiendo las particularidades de dicho servicio de modo que sea accesible para cualquier tipo de cliente.
- Cisco adoptó la SOA para asegurarse de que su experiencia de pedidos de productos fuera consistente en todos los productos.

### Ejemplos de Arquitecturas ESB.

- ESB de Synapse: es capaz de procesar un volumen grande de mensajes con un mínimo retardo y uso de recursos. Utiliza Axis2 como motor de servicios web.
- WS02 ESB: es un bus de servicio rápido ligero y flexible, es 100% software libre y distribuido bajo la licencia de Apache Software versión 2.0.



## MQTT Server.

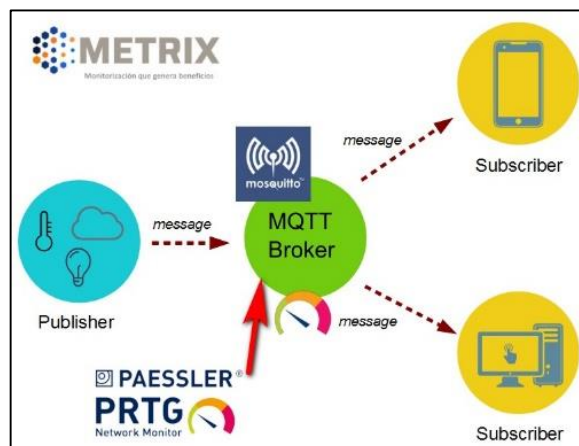
Son las siglas MQ Telemetry Transport fue conocido como Message Queuing Telemetry Transport. Es un protocolo de comunicación M2M, está orientado a la comunicación de sensores, debido a que consume muy poco ancho de banda y puede ser utilizado en la mayoría de los dispositivos empotrados con pocos recursos (CPU, RAM)

### Síntesis

Transporte de telemetría de cola de mensajes. es un protocolo que permite la comunicación, es utilizado en aplicaciones en tiempo real, con el fin de enviar información de manera rápida.

### Ejemplos de MQTT Server

- Aplicación Facebook messenger ya que hace uso de la topología de estrella con un nodo central que hace de servidor o "broker" con capacidad de 10,000 clientes
- Mosquitto: es un broker mqtt de código abierto es liviano y adecuado para su uso en todos los dispositivos.





## Rest-API

Una API (Interfaz de programación de aplicaciones) es un conjunto de requisiciones que permite la comunicación de datos entre aplicaciones. La API utiliza requisiciones HTTP responsables de las operaciones básicas necesarias para la manipulación de datos. Principales solicitudes:

- POST: crea datos en el servidor
- GET: lectura de datos en el host
- DELETE: borra la información
- PUT: registro de actualizaciones

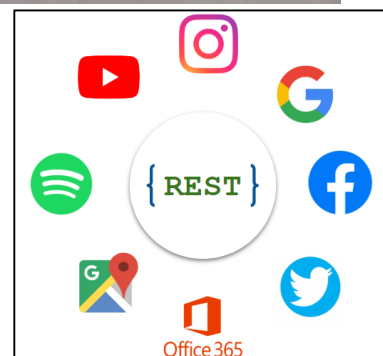
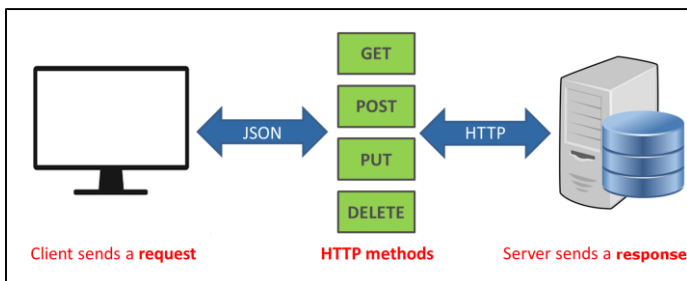
Rest, es la abreviación de Representational State Transfer, es un conjunto de restricciones que se utilizan para que las solicitudes HTTP cumplan con las directrices definidas en la arquitectura.

### Síntesis

Una API-REST es una aplicación web del lado de backend que proporciona un servicio en el cual se tiene una serie de rutas configuradas que hacen ciertas funcionalidades que es consumida mediante el protocolo HTTP.

### Ejemplo API REST

- Red Hat: es una solución centrada en las API que permite conectar las aplicaciones y transformar los datos, crear y organizar servicios, transmitir mensajes de forma instantánea, capturar datos modificados y gestionar las API.



## RPC - remote procedure call.

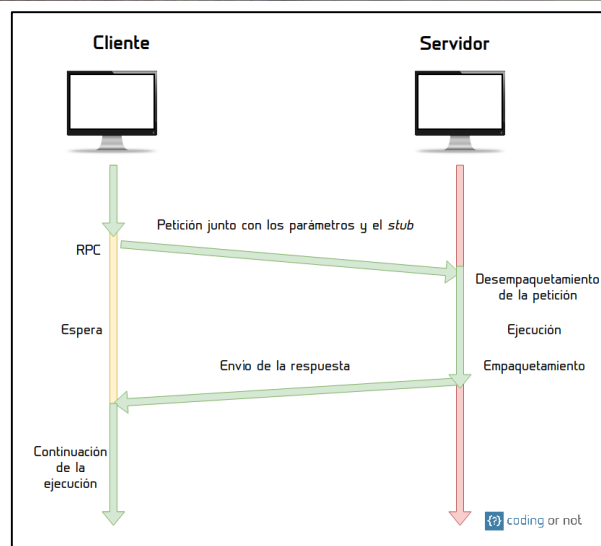
La llamada a procedimiento remoto (RPC) es un protocolo que un programa puede usar para solicitar un servicio de un programa ubicado en otra computadora en una red sin tener que comprender los detalles de la red. RPC se utiliza para llamar a otros procesos en los sistemas remotos como un sistema local. Una llamada a procedimiento también se conoce a veces como llamada a función o llamada subrutina. Usa el modelo cliente-servidor. El programa solicitante es un cliente y el programa proveedor de servicios es el servidor.

### Síntesis.

RPC es un protocolo que permite a un programa desde una computadora ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

### Ejemplos de software RPC

- Portmap: mapeador de puertos. Llamada de procedimiento remoto de computación en red abierta que se ejecuta en los nodos de red que brindan otros servicios OMC RPC.
- Network File System: Protocolo de nivel de aplicación. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local.



## REFERENCIAS BIBLIOGRAFICAS

- 4r Soluciones. (19 de Febrero de 2019). *¿Cómo medir la calidad en software?* Obtenido de 4r Soluciones: <https://www.4rsoluciones.com/blog/como-medir-la-calidad-en-software-2/>
- Architectural Pattern. (6 de Febrero de 2021). *Architectural Pattern*. Obtenido de Cio wiki: [https://cio-wiki.org/wiki/Architectural\\_Pattern](https://cio-wiki.org/wiki/Architectural_Pattern)
- Cisco. (s.f.). *Protocolos y estándares de red*. Obtenido de Cisco: <http://itroque.edu.mx/cisco/cisco1/course/module3/3.2.2.1/3.2.2.1.html>
- Cityclíc. (2020). *¿QUÉ ES SOA O ARQUITECTURA ORIENTADA A SERVICIOS?* Obtenido de Cityclíc: <https://www.ecityclíc.com/es/noticias/que-es-soa-o-arquitectura-orientada-a-servicios>
- FERNANDES, H. M. (08 de Julio de 2020). *¿Qué es un sistema/aplicación monolito/monolítico?* Obtenido de Marquesfernandes: <https://marquesfernandes.com/es/tecnologia-es/o-que-e-um-sistema-aplicacao-monolith-monolithic/>
- Gálvez, J. A. (2015). *¿Qué es MQTT?* Obtenido de Geeky theory: <https://geekytheory.com/que-es-mqtt>
- Goncalves, W. (21 de Agosto de 2019). *Qué son los stakeholders, qué tipos existen y de qué manera impactan a una empresa*. Obtenido de rockcontent: <https://rockcontent.com/es/blog/que-es-un-stakeholder/>
- IBM. (18 de julio de 2019). *ESB (Enterprise Service Bus)*. Obtenido de IBM: <https://www.ibm.com/mx-es/cloud/learn/esb>
- IBM. (2020). *Remote Procedure Call*. Obtenido de IBM: <https://www.ibm.com/docs/en/aix/7.1?topic=concepts-remote-procedure-call>
- Investopedia. (21 de Julio de 2021). *Enterprise Resource Planning (ERP)*. Obtenido de Investopedia: <https://www.investopedia.com/terms/e/erp.asp>
- JC Source Code. (1 de Octubre de 2019). *QUE SON LOS PATRONES ARQUITECTÓNICOS*. Obtenido de JC Source Code: <https://www.youtube.com/watch?v=cehDbpZviW4>
- Laura. (25 de Septiembre de 2017). *Qué es un ERP y para qué sirve - Definición de ERP*. Obtenido de Aplimedia: [https://www.youtube.com/watch?v=7\\_r7rGHmh1c](https://www.youtube.com/watch?v=7_r7rGHmh1c)
- Leiva, A. (8 de Marzo de 2016). *Patrones de diseño de software*. Obtenido de devexperto: <https://devexperto.com/patrones-de-diseno-software/>
- Microfocus. (2021). *¿Qué son las pruebas de rendimiento?* Obtenido de Microfocus: <https://www.microfocus.com/es-es/what-is/performance-testing>
- MicroTutoriales DC. (19 de Agosto de 2019). *#6 MicroPython - ¿Qué es MQTT? Configurar servidor CloudMQTT | ESP8266 y ESP32*. Obtenido de MicroTutoriales DC: <https://www.youtube.com/watch?v=pYKQJp-8UTA>
- Robles, V. (27 de Marzo de 2019). *¿Que es un API REST? ✓ ¿Para que sirve un API RESTful? ✨ ¿El futuro del Backend?*. Obtenido de Víctor Robles web: <https://www.youtube.com/watch?v=4fZKFQjkNw0>
- Sánchez, J. F. (02 de Julio de 2018). *Pruebas de rendimiento con JMeter. Ejemplos*



*básicos*. Obtenido de sdos: <https://www.sdos.es/blog/pruebas-de-rendimiento-con-jmeter-ejemplos-basicos>

Verdesoto, K. (Julio de 24 de 2015). *Criterios De Calidad En El Desarrollo De Software K.V*. Obtenido de Prezi: <https://prezi.com/xk6p4uitaevu/criterios-de-calidad-en-el-desarrollo-de-software-kv/>

Wikipedia. (15 de Julio de 2021). *Arquitectura de software*. Obtenido de Wikipedia: [https://es.wikipedia.org/wiki/Arquitectura\\_de\\_software](https://es.wikipedia.org/wiki/Arquitectura_de_software)

WordPress. (1 de Marzo de 2017). *SUITES DE PROTOCOLOS Y ESTÁNDARES DEL SECTOR DE REDES*. Obtenido de Wordpress: <https://interpolados.wordpress.com/2017/03/01/suites-de-protocolos-y-estandares-del-sector-de-redes/>