



El futuro digital  
es de todos

MinTIC

# Formador: Sergio Medina

» **BackEnd  
Conceptualización**

xxx

## Ciclo No.3 – Semana No.5



**Misión  
TIC2022**



El futuro digital  
es de todos

MinTIC



# Arquitectura de microservicios



En desarrollo de software existen diversos tipos de aplicaciones o productos software, desde aplicaciones tipo escritorio, móviles, embebidas y aplicaciones web. Además, cada una de ellas tiene características particulares en su arquitectura y las tecnologías utilizadas como base para su funcionamiento.

En este ciclo centraremos nuestra atención en un tipo de aplicación web para el ejercicio práctico.



xxx

Los modelos más usados en el desarrollo de aplicaciones web, son los denominados cliente-servidor, que separan lo que se conoce como backend y frontend. En la siguiente figura, se muestran algunas tecnologías utilizadas.



Adaptada de <https://ed.team/>





Para comprender el funcionamiento de las aplicaciones, hay que partir desde su arquitectura, donde diferenciamos en una visita de alto nivel, la forma como se va a estructurar el proyecto; Básicamente se pueden plantear dos enfoques, las arquitecturas monolíticas y las arquitecturas de microservicios, cuya diferencia radica en la forma de gestionar el acceso a los recursos del sistema, teniendo en cuenta las mismas capas, presentación o interfaz de usuario, lógica y acceso a datos o persistencia.

En una arquitectura monolítica, el grado de acoplamiento de una aplicación es elevado o llega a su máximo grado, considerando que el software quede estructurado de tal manera que todos los aspectos funcionales están sujetos a una misma lógica. Es decir, no existen módulos separados del sistema y además, todos los elementos están alojados en un único servidor.

A diferencia de lo planteado anteriormente, una arquitectura de microservicios dispone cada proceso como un elemento independiente por módulos, que pese a que siguen la base de funcionamiento de la arquitectura monolítica, no interfieren con el funcionamiento de otros módulos de servicios coexistentes que dan respuesta a una misma capa de presentación o interfaz gráfica, que para el cliente es casi invisible lo que está sucediendo en el servidor o servidores que atienden sus peticiones.

En resumen, las arquitecturas monolíticas son adecuadas para aplicaciones donde de manera holística se tienen definidas las funcionalidades, hay poca tendencia a escalabilidad, no requiere constante actualización y se requiere rigidez a la hora de realizar cambios, en búsqueda de un alto rendimiento del sistema. Esto, en algunos casos, es considerado su mayor desventaja.

Por el contrario, en las arquitecturas de microservicios, al tener separado y definido cada módulo de manera independiente, facilita la creación de nuevos servicios sin que se afecte el funcionamiento de módulos existentes, generando alta flexibilidad al cambio y la escalabilidad de un sistema; son ideales para aplicaciones de cambio y crecimiento constante.



El futuro digital  
es de todos

MinTIC



## ARQUITECTURA MONOLÍTICA

Interface  
de usuario

Lógica de  
negocios

Acceso a  
los datos



Base de  
Datos

## ARQUITECTURA DE MICROSERVICIOS

Interface  
de usuario

Microservicio

Microservicio

Microservicio

Microservicio

Microservicio



Base de  
Datos

Base de  
Datos

Base de  
Datos

Base de  
Datos

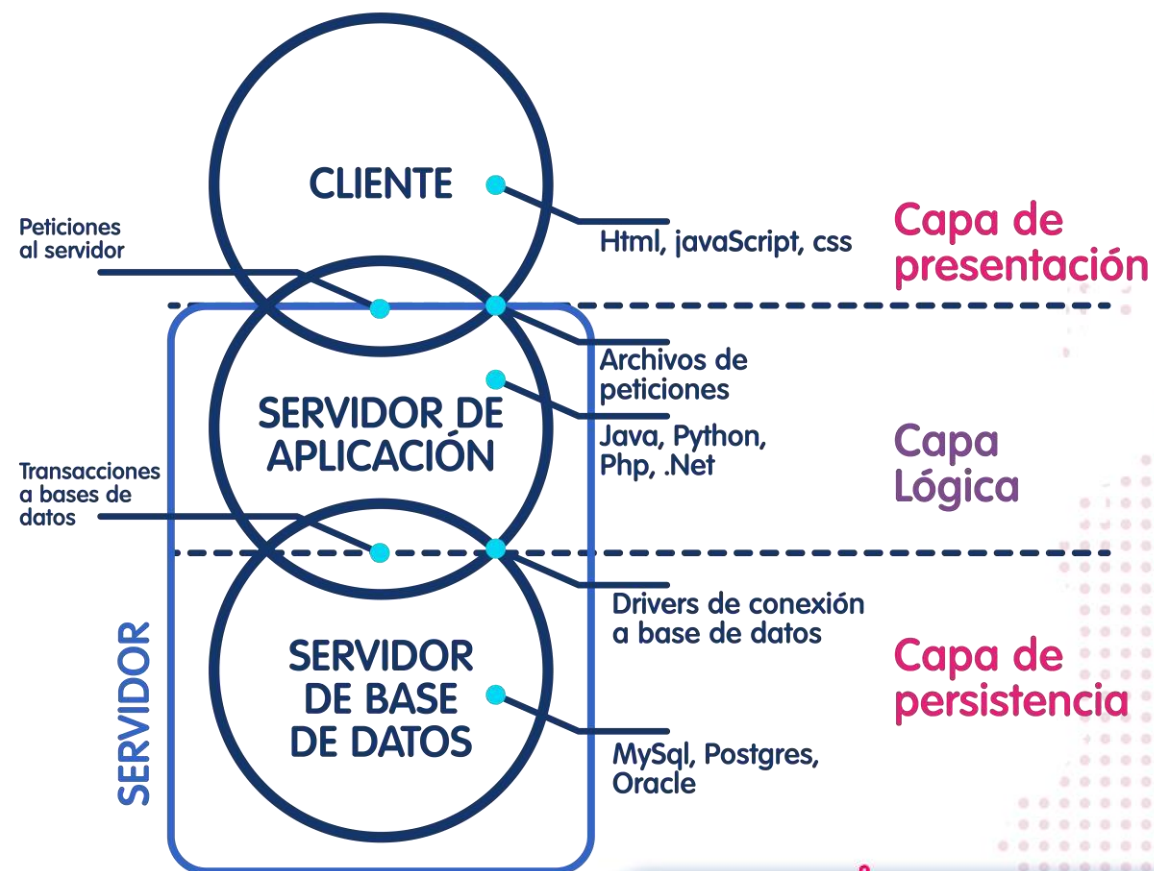
Adaptada de: <https://www.evaluandosoftware.com/que-es-la-arquitectura-de-microservicios/>



## » Configuración y creación del proyecto

×  
×  
×

Para el desarrollo de nuestra aplicación web, seguiremos este modelo por capas, donde se utilizará el lenguaje de programación JAVA como base para el backend y como servidor de aplicación apache tomcat, que nos permitirá alojar los archivos necesarios para la ejecución de nuestra aplicación y gestionar la conexión al servidor de base de datos, que como se muestra en la figura, puede coexistir junto al servidor de aplicación o bien podría estar separado y ser accedido de forma remota.



Modelo cliente servidor y capas de la aplicación web. Fuente: elaboración propia



**Capa de presentación:** Presenta al usuario final las Interfaces gráficas o GUI en las cuáles se puede hacer uso de los servicios desarrollados.

**Capa de negocio:** Describe y gestiona toda la parte lógica de la aplicación.

**Capa de persistencia:** Almacena y dispone la información necesaria para el funcionamiento de la aplicación.

Algo importante a destacar, es que entre los elementos cliente, servidor de aplicación y servidor de base de datos, existe un intermediario que facilita la comunicación.

Para el caso de la comunicación entre el cliente y el servidor, dispondremos por cada servicio de un archivo de peticiones que gestione los procesos solicitados por el cliente y permita utilizar los elementos de la capa lógica en el servidor de aplicación que den respuesta a lo solicitado por un usuario desde la capa de presentación. Por su parte, entre el servidor de aplicación y el servidor de base de datos, utilizaremos archivos de la capa de persistencia mediante un puente JDBC (Java Data Base Conector), que permitirá conectar la capa lógica a la base de datos para realizar operaciones de consulta o modificación.





El futuro digital  
es de todos

MinTIC

## » Desarrollo de proyecto Backend



Adaptado de: <https://www.tecnova.cl/2020/02/28/detras-de-toda-gran-app-hay-un-gran-mobile-backend/>





Para el desarrollo del proyecto en términos de responsabilidades backend se deben tener en cuenta aspectos como:

- Disposición de archivos para la aplicación (Imágenes, logos, documentos...).
- Gestión de acceso a rutas de archivos dispuestos en el servidor.
- Parametrizar la aplicación para conexiones a bases de datos, y conexiones remotas si las requiere.
- Control de la seguridad en términos de privacidad de los datos sensibles dispuestos en la base de datos.
- Desarrollo de piezas de software que permitan al sistema crecer de manera modular y estandarizada, para ello se recomienda la aplicación de paradigmas de programación como el orientado a objetos.
- Mantener actualizados diagramas UML, Diagramas entidad Relación y cualquier otro diagrama que permita documentar el proceso evolutivo del software.
- Atender a los procesos misionales de la organización o contexto donde se esté trabajando para que en lo posible se pueda tener respuestas rápidas a los requerimientos, teniendo en cuenta que el contexto cambia mucho más rápido que el dominio del sistema.



Como ejercicio práctico para esta sección, se espera desarrollar funcionalidades de configuración del proyecto conforme a los requerimientos planteados en las lecciones anteriores, donde se logre tener una comunicación básica con base de datos mediante un CRUD, para a lo menos uno de los servicios o funcionalidades esperadas en el alcance total del proyecto.

Sigla CRUD	Operación
<b>C</b> reate	Creación de un nuevo registro en base de datos
<b>R</b> ead	Lectura de registros de base de datos
<b>U</b> pdate	Actualización de registros de base de datos
<b>D</b> elete	Eliminación de registros de base de datos



## » Elementos a recordar para trabajar con Java

✕ A modo de revisar que contemos con las herramientas necesarias, es importante diferenciar entre lo que son JDK, JRE, JVM e IDE  
✕ además de los conceptos básicos del lenguaje.  
✕

### Java Development Kit

- Java Runtime Environment
- Java Virtual Machine
- Integrated Development Environment
- <https://javadesdecero.es/fundamentos/>

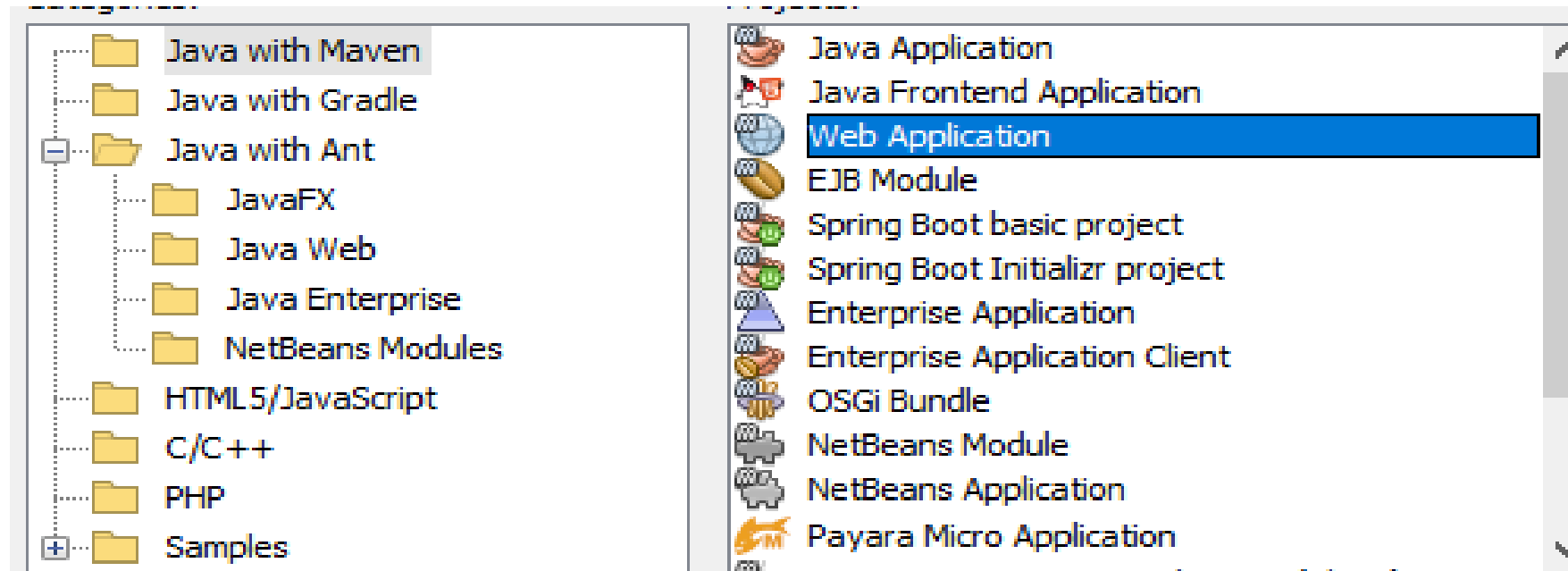
A continuación, se deja una lectura sugerida y enlaces de descarga de ser necesario:

- <https://javadesdecero.es/fundamentos/diferencias-jdk-jre-jvm/>
- <https://javadesdecero.es/fundamentos/jre-java-runtime-environment/>
- <https://www.oracle.com/co/java/technologies/javase-jre8-downloads.html>
- <https://www.oracle.com/co/java/technologies/javase/jdk8-downloads.html>

# Creación del proyecto

Dependiendo del tipo de arquitectura y de tecnologías empleadas, la integración con el servidor puede variar la forma de la integración, para el ejemplo de referencia se pueden utilizar Java Web Application Tipo Maven o Java with ant-Java Web si se quiere manejar el control de dependencias sin XML.

## Java Web Application







Categories:

- Java with Maven
- Java with Gradle
- Java with Ant
  - JavaFX
  - Java Web
  - Java Enterprise
  - NetBeans Modules
- HTML5/JavaScript
- C/C++
- PHP
- Samples

Projects:

- Web Application
- Web Application with Existing Sources
- webfreeform.xml

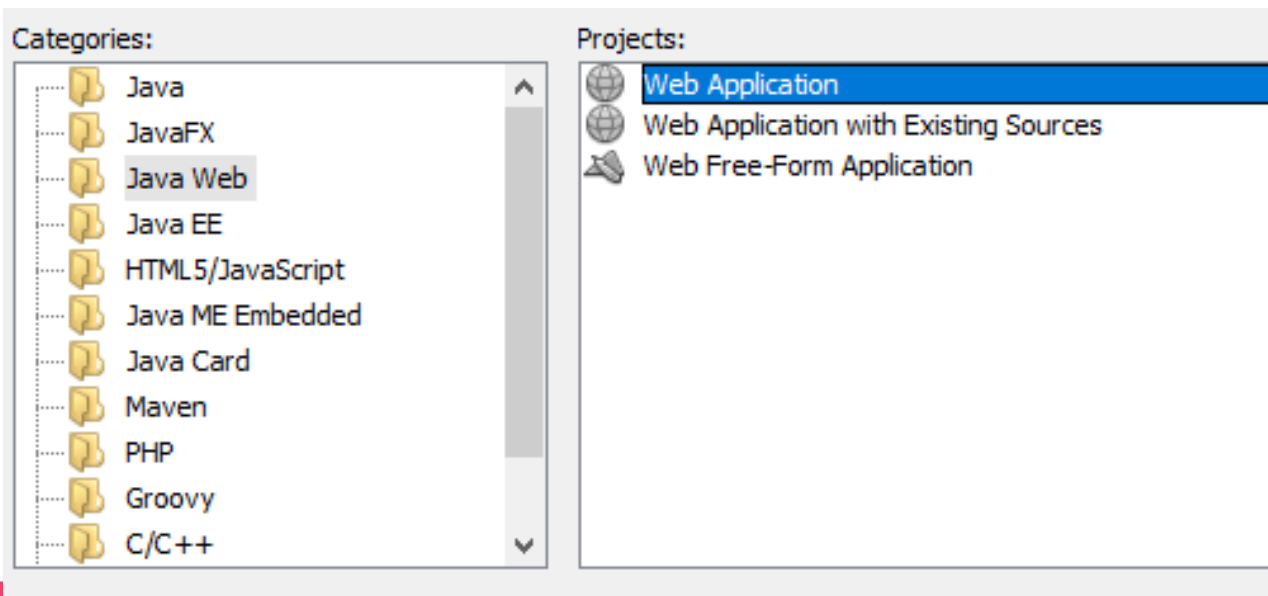
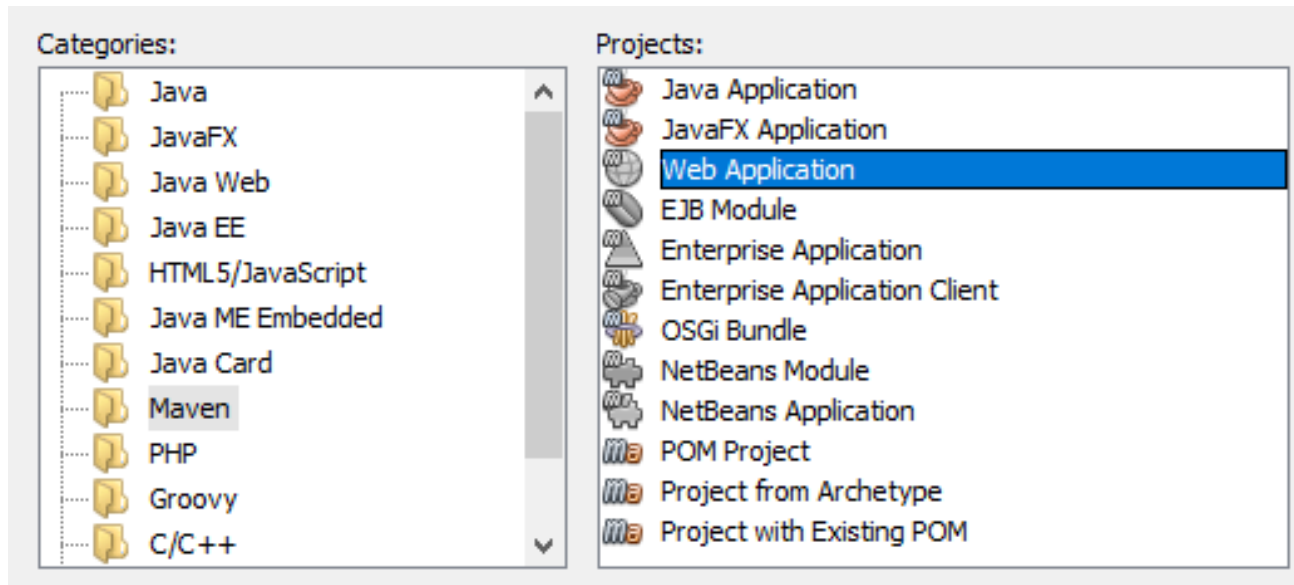
Si se cuenta con versiones anteriores como netbeans 8.1 u 8.2, se pueden integrar al aplicación mediante los proyectos de categoría Maven-Web Application y Java Web- Web Application.



El futuro digital  
es de todos



## Java Web- Web Application





## Configuración del proyecto a nivel de backend



Para esta parte del proceso, se debe configurar el proyecto solo nivel de los archivos del lado del servidor, en el la siguiente sección se agregaran elementos del lado del cliente.

Los archivos de mínimos de configuración del proyecto para el backend son los siguientes:

- Paquete de clases con nombre `restful.model`
  - Clase que represente el objeto para el cual se va a realizar el CRUD, clase de conexión a BD y clases de apoyo
- Paquete de clases con nombre `restful.resource`
  - Clase con el manejo de los recursos GET, POST, PUT y DELETE (Ejecutan servicios WEB) ( Son consumidos desde el Frontend)
- Paquete de clases con nombre `restful.service`
  - Clase con los servicios WEB (Operaciones SQL sobre la Base de Datos)



El futuro digital  
es de todos

MinTIC

# Formador: Sergio Medina

» **BackEnd  
Conceptualización**

xxx

## Ciclo No.3 – Semana No.5



Misión  
TIC2022