

Back end nivel intermedio

Programación en Java



En esta experiencia educativa aprenderás algunas de las principales características del lenguaje de programación Java. Algunas de estas características son las siguientes: es independiente de la plataforma; esto lo hace a través de bytecodes que son interpretados por la Java Virtual Machine y permite que se pueda ejecutar prácticamente en cualquier plataforma, otras de las bondades de este lenguaje, es orientado a objetos, sencillo, seguro, no depende de alguna arquitectura para funcionar, es portable y muy robusto, además de ser un lenguaje de alto rendimiento y que puede ejecutarse en arquitecturas distribuidas.

Este lenguaje ya tiene más de 20 años estando entre los más utilizados en la industria gracias a que permite desarrollar software para cualquier propósito, desde programas para escritorio, aplicaciones web, aplicaciones distribuidas, pasando por los microservicios y aplicaciones móviles.



Java es un lenguaje desarrollado por Sun Microsystems, empresa que posteriormente fue adquirida por Oracle. La sintaxis de Java es parecida a la de C o C++, solamente que Java agrega características de otros lenguajes como extensiones, por ejemplo, la gestión de hilos. Java es un lenguaje orientado a objetos, por lo cual; es necesario que tener claros los conceptos de la programación orientada a objetos, recuerda que es un paradigma para desarrollar software que trata de imitar la realidad tomando algunos conceptos esenciales de ella.

El primero de ellos es el concepto de **objeto**, del cual toma la identidad (nombre), el estado (características) y el comportamiento. Cuando se crea una copia de ese objeto se llama instancia. Dichos objetos son creados desde clases, que vendrían a ser como los moldes para cada objeto.

Otros conceptos son la **abstracción** y **encapsulación**, que están ligados al diseño de los programas. Dichos conceptos están relacionados con las tareas que los objetos pueden o deben hacer.

Y finalmente los conceptos de **herencia** y **polimorfismo**, permiten que los objetos creados tengan algunas características de sus padres, pero, además, puedan tener ellos mismos algunas características diferentes.

Entrando en materia, la programación en Java tiene algunas cosas inamovibles, la primera de ellas es la ejecución:

Para que puedas llevar a cabo una ejecución de una aplicación en Java debe existir una clase que tenga el método “main”.

```
public static void main(String[] args)
{
    ... // Código del método
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Dentro de este método deberás colocar tu código Java.

Clases

En Java siempre se van a utilizar clases, ya sea que nosotros las desarrollemos o que tomemos clases ya hechas y las importemos.

```
class ClaseJava
{
...
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Variables

Son espacios de memoria en los que se guarda información. Para la Programación Orientada a Objetos son las características de la Clase. Para nombrar las variables es conveniente que uses sustantivos que sean descriptivos al significado. Si están formados por más de una palabra; la primera palabra debe comenzar con minúscula y la segunda palabra deberá comenzar con la primera letra mayúscula. Java al ser un lenguaje altamente tipado, necesita que se definan el tipo de dato en cada una de las variables declaradas.

```
int edad;
String apellidos;
String fechaNacimiento;
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Métodos

Son las acciones que puede realizar la clase y pueden devolver cualquier tipo de datos, además que también pueden recibir parámetros (solo los métodos tipo void pueden no devolver algún valor y pueden o no recibir parámetros). Al igual que las variables, se escriben con la primera palabra en minúsculas y el resto comenzando por mayúsculas Los métodos se componen de tipo de datos que devuelven, el nombre del método, y luego los argumentos entre paréntesis.

```
void saludar(String mensaje)
{
...
// Código del método
}
double multiplicar(double... numeros){
//Número variable de argumentos
//Se accede a ellos como a un vector:
//numeros[0], numeros[1], ...
}||
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Constructores

Una de las responsabilidades de cualquier clase es autoconstruirse, para esto, se utilizan a los constructores que se llaman igual que la clase, estos deberán ser públicos y no tienen modificador del tipo que devuelven.

```
//Un ejemplo simple de constructor
class MiClase {
    int x;

    public MiClase() {
        x = 10;
    }
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

La siguiente imagen, es un ejemplo de constructores sin parámetros y constructores con paso de parámetros:

Se declara la clase con dos constructores:

```
/**
 * Declaracion de la clase Fecha
 */
public class Fecha {
    // Atributos o variables miembro
    private int dia;
    private int mes;
    private int anio;

    /**
     * Constructor 1
     * Asigna los valores a los atributos
     * dia, mes y anio respectivamente
     */
    public Fecha() {
        this.dia = 15;
        this.mes = 9;
        this.anio = 2000;
    }

    /**
     * Constructor 2
     * @param dia el dia del mes a almacenar
     * @param mes el mes del anho a almacenar
     * @param anio el anio a almacenar
     */
    public Fecha(int dia, int mes, int anio) {
        this.dia = dia;
        this.mes = mes;
        this.anio = anio;
    }

    public String toString() {
        return this.dia + "/" + this.mes + "/" + this.anio;
    }
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Luego se usa o implementa desde la siguiente clase:

```
/**  
 * Ejemplo de uso de la clase Fecha  
 */  
public class PruebaFecha {  
    Run | Debug  
    public static void main(String[] args) {  
        Fecha origen = new Fecha();  
        Fecha actual = new Fecha(dia: 16, mes: 2, anio: 2009);  
  
        System.out.println("Primera fecha: " + origen.toString());  
        System.out.println("Segunda fecha: " + actual.toString());  
    }  
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Arrays en Java

En Java, un arreglo (array) es una estructura de datos la cual permite almacenar más de una variable de un mismo tipo. Por ejemplo, si se tienen números enteros y su capacidad es 10, solo podrán almacenarse esa cantidad de elementos de ese tipo. La declaración de un array en Java se puede indicar de la siguiente manera:

```
tipo_dedato [] nombre_variable; //tambien puede declararse “tipo_dedato  
nombre_variable[];
```

Ejemplos:

```
String array[]; //donde array es un arreglo que guardara variables tipo Strings  
double array2[]; //donde array2 es un arreglo que guardara variables tipo double  
int [] array3; //donde array3 es un arreglo que guardara variables tipo int(numericas)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Los agarra y comienzan a realizar una enumeración desde el elemento cero hasta el último elemento o en su defecto la posición -1.

Para crear un array y colocar datos en Java puedes realizarlo de la siguiente manera:

```
// Array de 5 elementos.  
char array5[] = {'a','b','c','d','e'};
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

En este ejemplo la posición 0 del array es la letra a, la posición 1 es la letra b y así sucesivamente.

Para acceder a los valores por posición, se coloca entre corchetes la posición a la que quieras acceder, por ejemplo:

```
System.out.println("Elemento 2 del arreglo array5: " + array5[2]);
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Nos da como resultado:

Elemento 2 del arreglo array5: c

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Hasta este momento solo has visto como visualizar un dato a la vez, pero si queremos visualizar todos los datos que contiene un array, lo puedes hacer con un bucle for. Usaremos una de las propiedades de los arrays la cual es longitud(length). Esta propiedad nos permite saber cuántos elementos contiene un array.

```
int[] intArray = {1, 2, 3, 5};

for( int i=0; i< intArray.length; i++){
    System.out.println(intArray[i]);
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Este bucle recorrerá el array y nos dará como resultado lo siguiente.

```
D:\jdk-16.0.1\bin\java.exe "-javaagent:D:\Program Files\IntelliJ IDEA 2021.3.3\lib\idea_r
1
2
3
5

Process finished with exit code 0
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Dentro del array también existen las “matrices” estas no son más que array con subíndices o arrays de dos niveles o de dos dimensiones.

Las matrices se declaran de la siguiente manera:

```
int matriz[][] = {{1,2,3,4}, {5,6,7,8,9}};
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

En este ejemplo se tienen dos niveles en un mismo array para acceder a un elemento, por ejemplo, queremos imprimir el número 6 debemos entrar al segundo nivel del array, el cual es 1, ya que recuerda que los array comienzan desde el número 0.

Una vez que estemos en el nivel 1 la posición del número 6 es 1 entonces quedaría de la siguiente manera:

```
System.out.println(matriz[1][1]);
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Arquitecturas Java EE

Java EE provee a los desarrolladores un modelo de aplicación, el cual; permite el desarrollo de aplicaciones multicapa que dejan asegurar la escalabilidad, accesibilidad y facilidad de gestión.

En esta arquitectura de las aplicaciones multinivel se dividen el trabajo a realizar durante la implementación en dos.

Por una parte, la lógica de presentación y de negocio; utiliza un modelo de programación distribuido en distintas capas, para lograr que se divida en varios componentes dependiendo de su funcionalidad y despliegue, por otra parte, están los servicios que ofrece J2EE.

Según IBM (2021), los niveles o capas son:

- Capa del cliente. En este nivel se ejecutan las llamadas al servidor Java EE, el cual se encarga de procesar y regresar una respuesta al cliente.
- Capa web. Es la responsable de manejo del flujo de la aplicación, obtener los resultados de la capa de negocio y se encarga de administrar las sesiones del cliente.

- Capa de negocio. Como tal, es la encargada de la lógica de la aplicación. Aquí se utilizan componentes de Java EE como lo es el EJBs, esta capa se comunica con el acceso a datos, sin embargo, cabe destacar que esta capa no debe tener conocimiento alguno sobre la capa de acceso a datos, debido a que las modificaciones no deben de afectar a las otras capas.
- Capa de datos. Conocida también como el Sistema de Información Empresarial (EIS por sus siglas en inglés) es la encargada de la persistencia de los datos a través de algún mecanismo de almacenamiento como: base de datos, ERP, motores transaccionales, etc. En esta capa no se almacena ninguna lógica de trabajo.

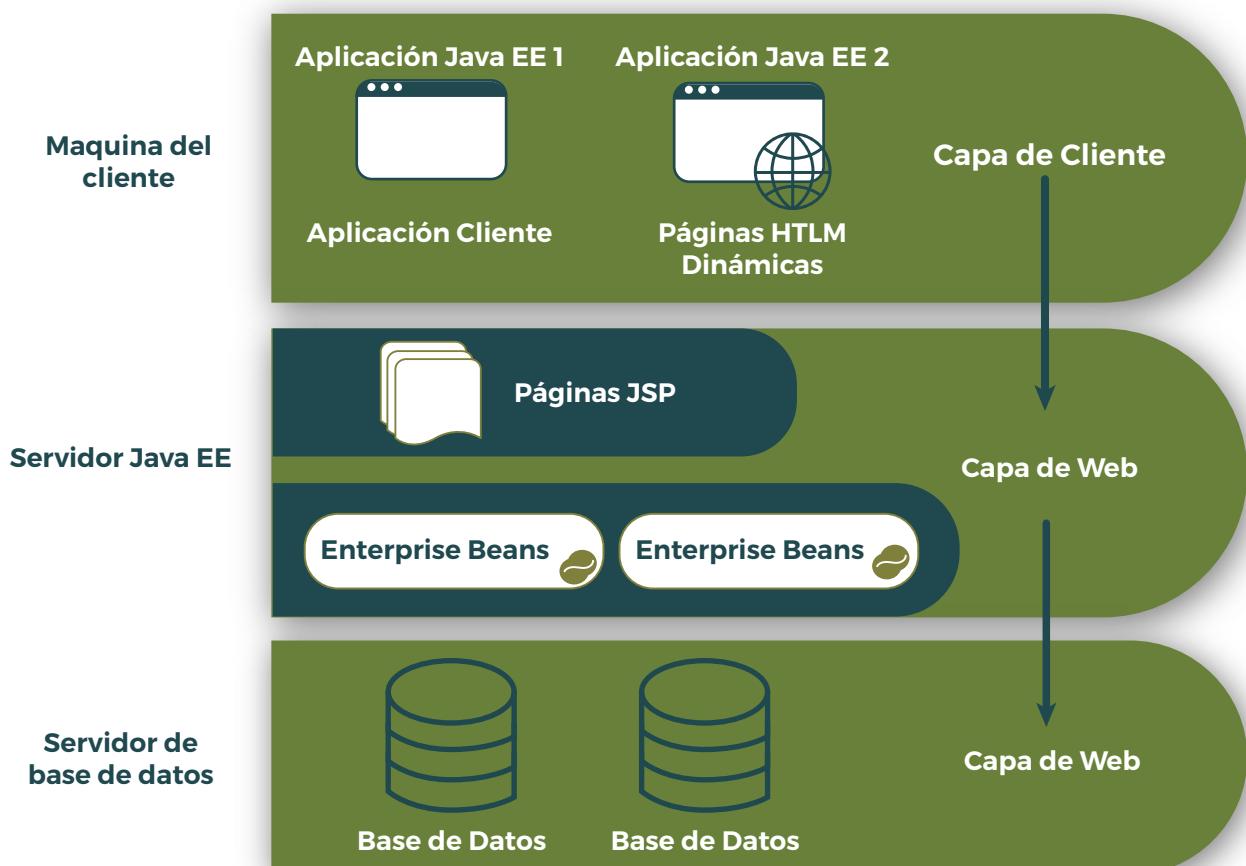


Figura 1. Arquitectura Java EE.

Microservicios JAVA y api REST

Hoy en día, cuando se tiene que desarrollar una aplicación nueva, durante las etapas tempranas del ciclo de vida, se tiene que elegir la arquitectura que se desea utilizar, como lo explica Romero (2021), se tiene la opción de utilizar un modelo monolítico, en el que todos los servicios que requiere el sistema son proporcionados por un único “paquete” o utilizar el modelo de microservicios en el que ese “paquete” se divide en paquetes que son más pequeños e independientes entre sí, pero que pueden comunicarse entre ellos.

Explicación

Back end nivel intermedio

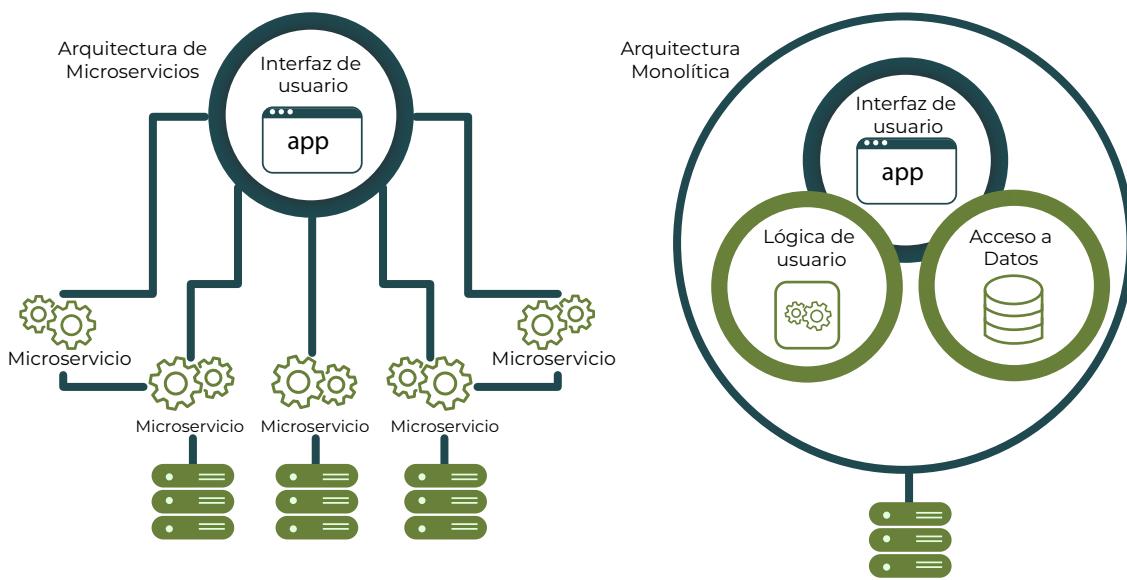


Figura 2. Diferencia entre arquitectura monolítica vs arquitectura de microservicios.

Oracle (s.f.), nos advierte que para desarrollar un microservicio se requiere de múltiples herramientas y más si es que se desea que el microservicio pueda ser usado en la nube. Por ejemplo, en el siguiente diagrama se puede ver la arquitectura de una aplicación que tiene varios microservicios:

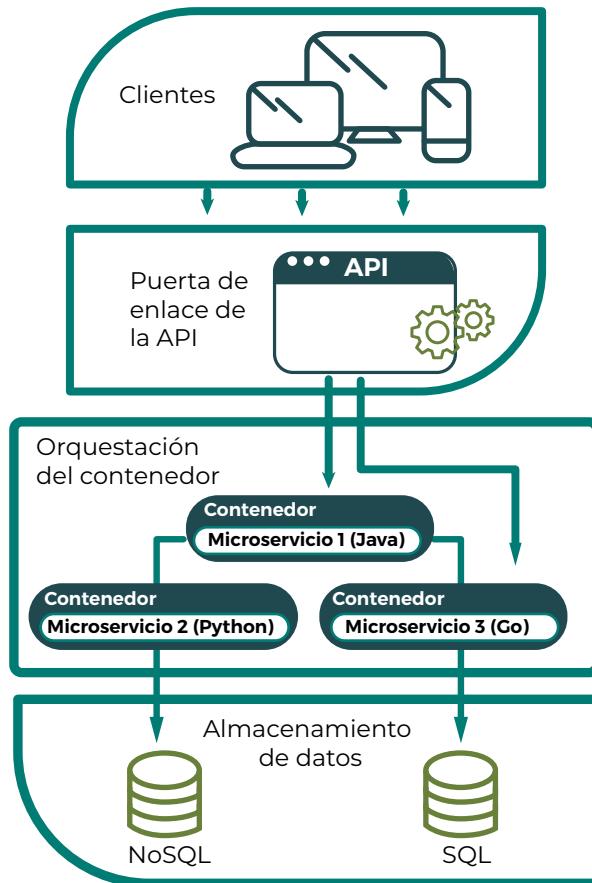


Figura 3. Arquitectura de aplicación con varios microservicios.

Oracle (s.f.), recomienda que, para desarrollar microservicios en Java, se pueden utilizar las herramientas “Helidon” como servidor y “Docker” para la gestión de contenedores. Un contenedor es como una máquina virtual pero no es lo mismo. Este permite ejecutar aplicaciones en su propio espacio de memoria al mismo tiempo que mantiene el acceso a recursos comunes como los sistemas de archivos.

Como se puede observar en el esquema anterior, los microservicios se conectan entre si a través de un programa de aplicación (API por sus siglas en inglés). Existen varios tipos de API, API Privadas, API Pùblicas, API de SOA y API REST. De todas estas, se ahondará un poco por las API REST (Representational State Transfer), por ser las que más se utilizan en el desarrollo de software hoy en día.

Según RedHat (2020), REST no es un protocolo, ni un estándar, sino un conjunto de límites de arquitectura, esto es, cuando un cliente envía una solicitud a través de una API de RESTful, lo que se está realizando, es que se está transfiriendo una representación del estado del recurso requerido a quien lo haya solicitado. Esta información se entrega a través de HTTP en alguno de los formatos acordados (JSON, HTML, XML, etc.) en un texto sin formato. Una definición más sencilla sería que REST es cualquier interfaz para realizar una operación u obtener datos entre sistemas que utilicen HTTP en cualquier de los formatos permitidos, y es una alternativa ante otros protocolos gracias a que su complejidad es menor.

Para ejemplificar los conceptos de RESTful, se tomará una agenda de empleados. La aplicación está compuesta de dos partes fundamentales para su funcionamiento que son:

- Un back end de microservicios escritos en Java.
- Un cliente de servicio en HTML, CSS y JavaScript.

Diagrama de infraestructura de la aplicación.

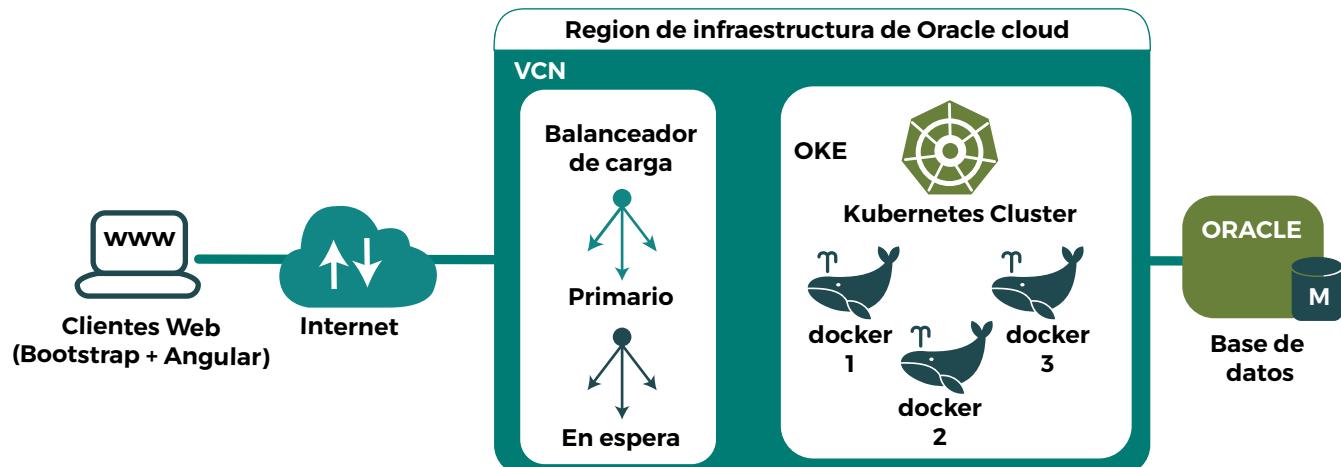


Figura 4. Infraestructura de la aplicación.

El cliente accederá a la app por medio de la red.

El back end de la aplicación será un conjunto de contenedores que ejecutan distintos servicios en cluster de kubernetes, el cual se encarga de administrar la carga. La aplicación consume una base de datos de Oracle para la persistencia. Esta base de datos no guarda ningún estado dentro del clúster de los microservicios.

El back end de la aplicación implementa los servicios CRUD (create, read, update, delete) y se implementa en Java a través del servidor Helidon.

Estos podrían ser algunas de las rutas de acceso de la aplicación a puntos finales

Ruta de acceso	Método	Descripción
/	GET	Obtener todos los empleados.
/apellido/{nombre}	GET	Busque todos los campos apellido para el valor proporcionado en "nombre".
{id}	POST	Agregar un nuevo empleado.
{id}	UPDATE	Actualiza los datos de un empleado especificado por el ID.
{id}	DELETE	Suprimir un empleado conforme al ID.

Tabla 1. Rutas de acceso de aplicación CRUD.

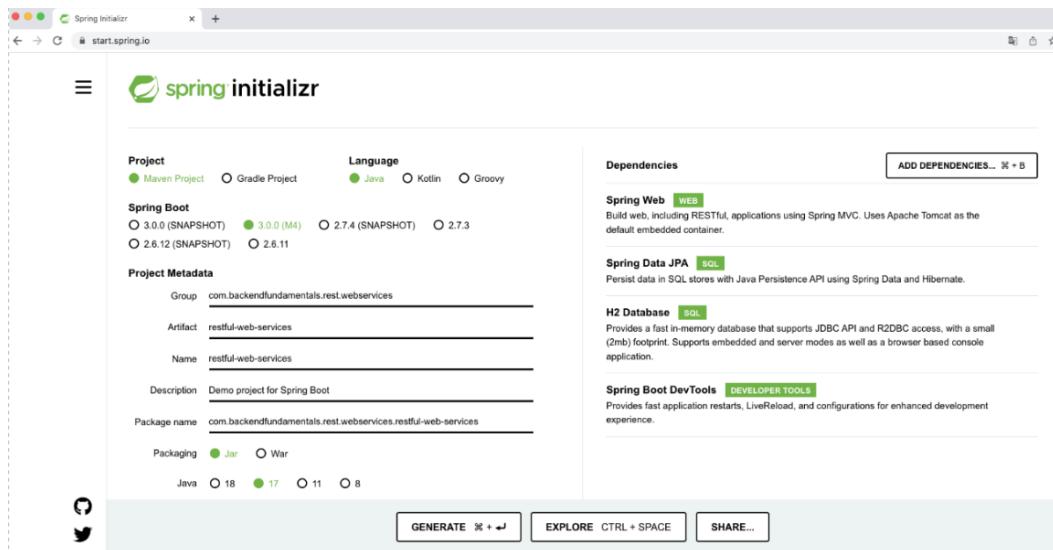
Para crear una aplicación restful desde 0 en java, se requiere crear el proyecto desde la herramienta Spring Initializr en el sitio web:



Start spring. (s.f.). *Spring initializr*. Recuperado de <https://start.spring.io>

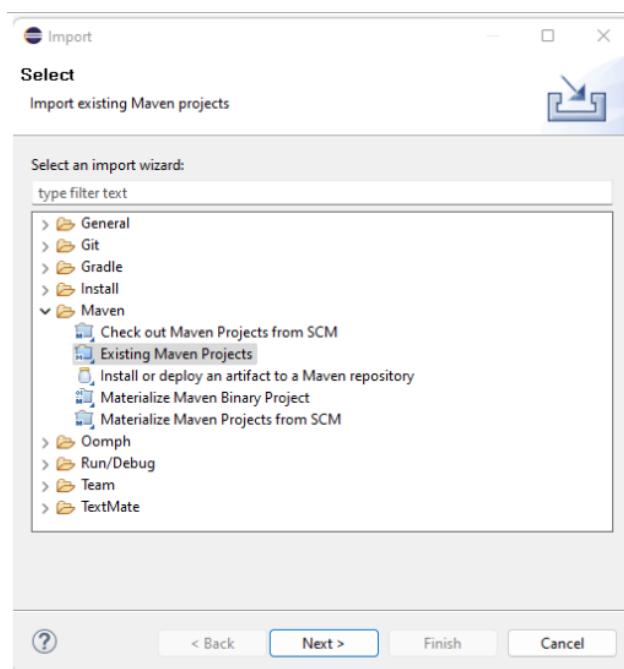
Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Configurar conforme a las siguientes opciones:



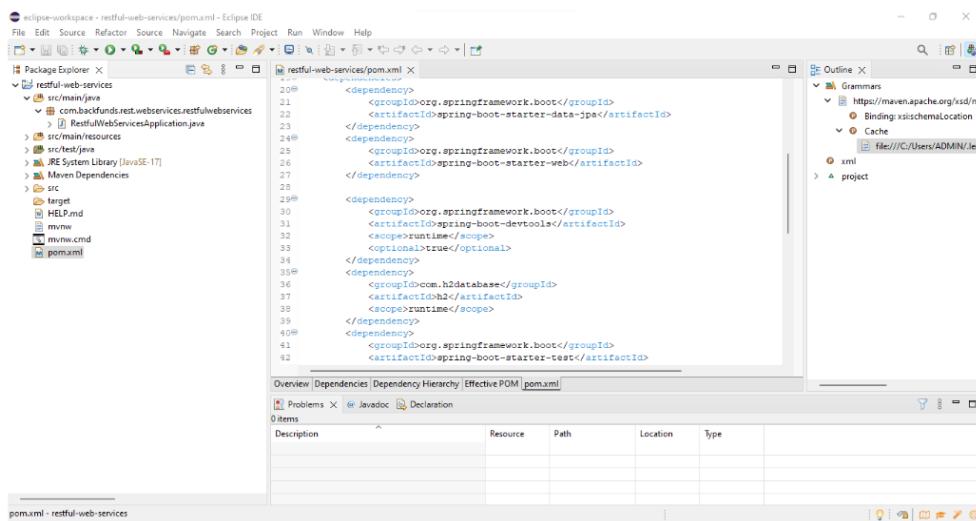
Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Una vez que se ha creado el proyecto, se tiene que importar como un proyecto existente de Maven usando el IDE Eclipse.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Una vez que hayas importado el proyecto, podrás visualizar todos los archivos incluidos en el proyecto, entre ellos el archivo: pom.xml, el cual contiene toda la configuración del proyecto.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para validar que el proyecto se importó correctamente, es necesario ejecutar la clase: RestfulWebServicesApplication.java como una aplicación Java.

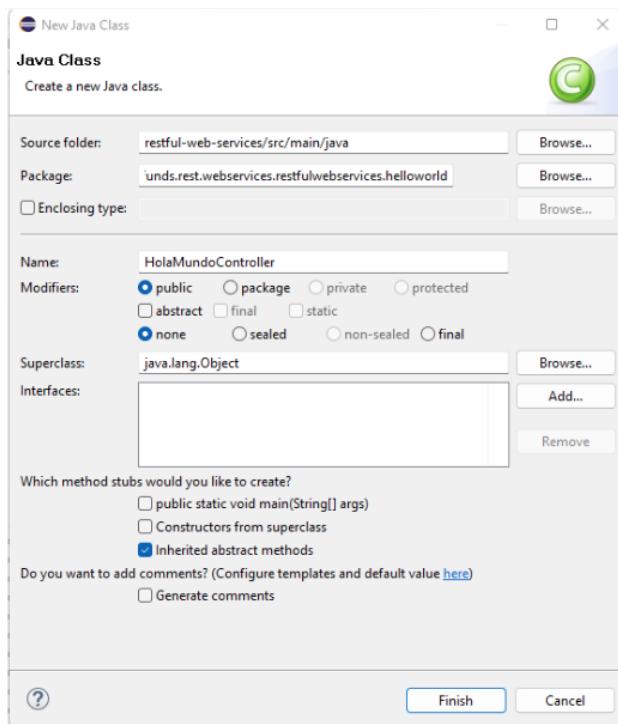
```

2022-09-20T00:37:07.114-05:00 INFO 19632 --- [ restartedMain] c.b.r.w.r.RestfulWebServicesApplication : Starting RestfulWebServicesApplication using Java 17.0...
2022-09-20T00:37:07.122-05:00 INFO 19632 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : No active profile set, falling back to 1 default profile: 'tomcat'
2022-09-20T00:37:07.270-05:00 INFO 19632 --- [ restartedMain] e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.restart.enabled=false' for additional web related logging consider setting the 'spring.devtools.restart.enabled=false' property
2022-09-20T00:37:08.418-05:00 INFO 19632 --- [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode
2022-09-20T00:37:10.000-05:00 INFO 19632 --- [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 15 ms. Found 0 repository interfaces
2022-09-20T00:37:09.587-05:00 INFO 19632 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-09-20T00:37:09.589-05:00 INFO 19632 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-09-20T00:37:09.596-05:00 INFO 19632 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.0.22]
2022-09-20T00:37:09.704-05:00 INFO 19632 --- [ restartedMain] o.a.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext: initialization completed in 1 ms
2022-09-20T00:37:09.709-05:00 INFO 19632 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-09-20T00:37:09.757-05:00 INFO 19632 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:...
2022-09-20T00:37:10.000-05:00 INFO 19632 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-09-20T00:37:10.015-05:00 INFO 19632 --- [ restartedMain] o.s.b.w.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'. Database available.
2022-09-20T00:37:10.347-05:00 INFO 19632 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2022-09-20T00:37:10.414-05:00 INFO 19632 --- [ restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.1.1.Final
2022-09-20T00:37:10.664-05:00 WARN 19632 --- [ restartedMain] org.hibernate.orm.deprecation : HHH00000021: Encountered deprecated setting [javax.pers...
2022-09-20T00:37:10.955-05:00 INFO 19632 --- [ restartedMain] org.hibernate.SQLialect : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
2022-09-20T00:37:11.277-05:00 INFO 19632 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.jta.platform.internal.JtaPlatformImpl]
2022-09-20T00:37:11.290-05:00 INFO 19632 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit.
2022-09-20T00:37:11.426-05:00 WARN 19632 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during a view's rendering. If this causes issues, either disable spring.jpa.open-in-view or configuration...
2022-09-20T00:37:12.035-05:00 INFO 19632 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2022-09-20T00:37:12.128-05:00 INFO 19632 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path
2022-09-20T00:37:12.158-05:00 INFO 19632 --- [ restartedMain] c.b.r.w.r.RestfulWebServicesApplication : Started RestfulWebServicesApplication in 5.484 seconds

```

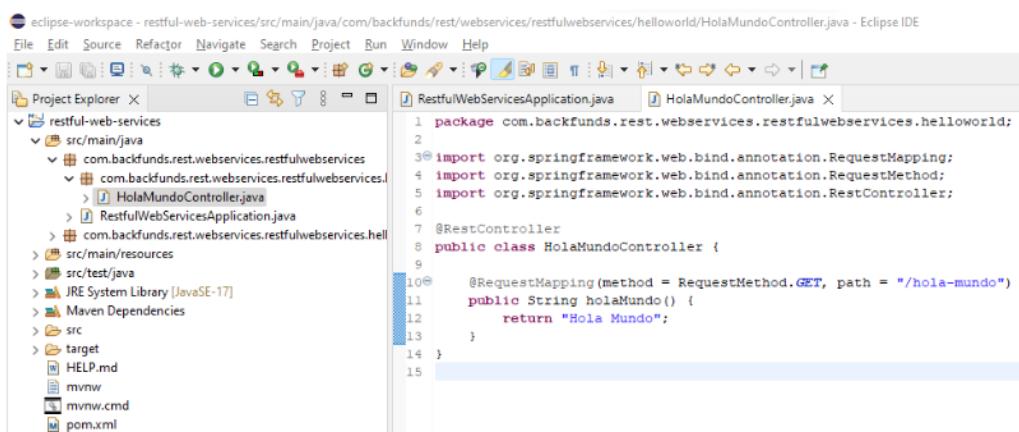
Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Ahora crearemos una clase para probar el servicio, para ello; es necesario hacer clic en el paquete “restful web service” y agregar una nueva clase.



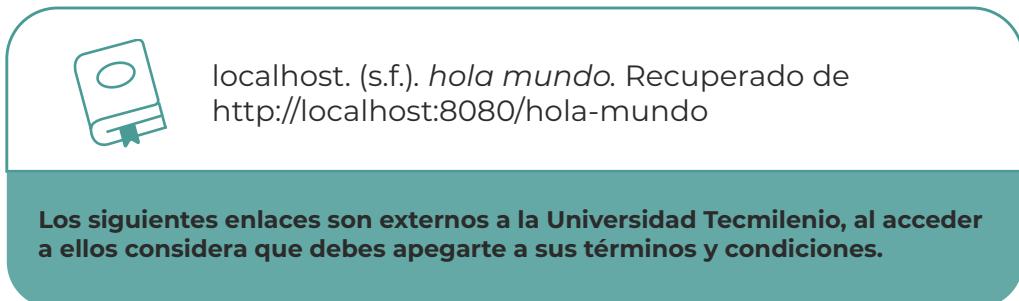
Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Esta clase será el controlador del recurso, en el cual se requiere crear un método que regresará la cadena hola mundo, que es una clase común y corriente, excepto porque tiene unas anotaciones: la anotación @RestController indica que es un controlador Rest y que debe tratarlo como tal cuando se ejecute, el @RequestMapping que contiene el método, por el cual se invoca a la API rest y el path en donde se encuentra publicado.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

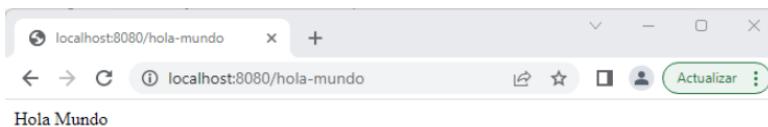
Una vez que se ha compilado, se puede acceder a la API desde el navegador en la url:



A screenshot of a web browser window. The address bar shows 'localhost:8080/hola-mundo'. The main content area displays the text 'Hola Mundo'.

localhost. (s.f.). *hola mundo*. Recuperado de
http://localhost:8080/hola-mundo

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

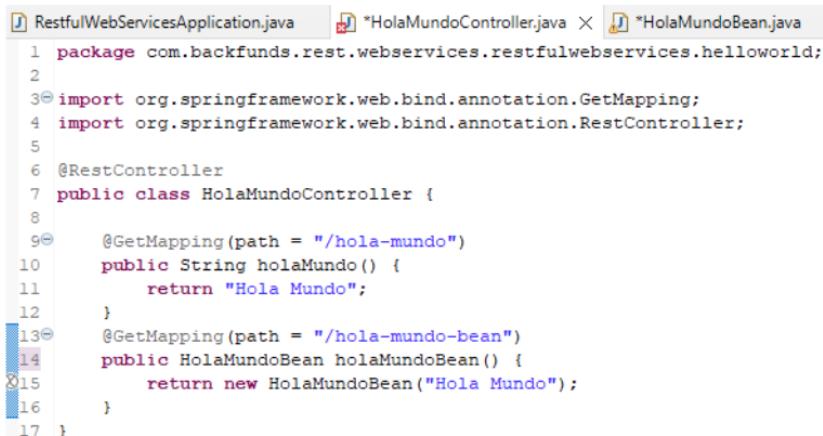


Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Hasta ahora se creó una API que regresa una cadena, sin embargo, típicamente las APIs regresan un mensaje en JSON, para ello, es necesario modificar la API REST para que regrese un **bean** (se utilizan para encapsular varios objetos en uno solo) de Java.

Los cambios que se tienen que realizar son 2:

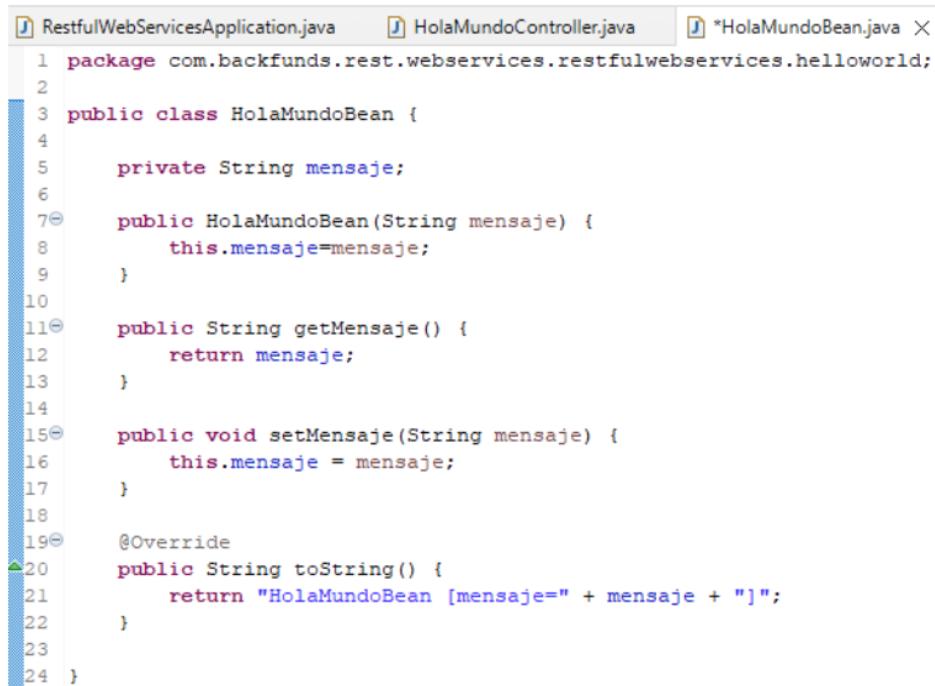
Agregar un **bean** a la clase creada.



```
RestfulWebServicesApplication.java *HolaMundoController.java *HolaMundoBean.java
1 package com.backfunds.rest.webservices.restfulwebservices.helloworld;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HolaMundoController {
8
9     @GetMapping(path = "/hola-mundo")
10    public String holaMundo() {
11        return "Hola Mundo";
12    }
13    @GetMapping(path = "/hola-mundo-bean")
14    public HolaMundoBean holaMundoBean() {
15        return new HolaMundoBean("Hola Mundo");
16    }
17 }
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Después, crear una nueva clase con el bean, con un constructor que recibe como parámetro el mensaje.



```
RestfulWebServicesApplication.java HolaMundoController.java *HolaMundoBean.java X
1 package com.backfunds.rest.webservices.restfulwebservices.helloworld;
2
3 public class HolaMundoBean {
4
5     private String mensaje;
6
7     public HolaMundoBean(String mensaje) {
8         this.mensaje=mensaje;
9     }
10
11     public String getMensaje() {
12         return mensaje;
13     }
14
15     public void setMensaje(String mensaje) {
16         this.mensaje = mensaje;
17     }
18
19     @Override
20     public String toString() {
21         return "HolaMundoBean [mensaje=" + mensaje + "]";
22     }
23
24 }
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Y la salida sería la siguiente:



```
192.168.1.13:8080/hola-mundo-bean
{
    "mensaje": "Hola Mundo"
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Otra manera que se puede consumir la API es a través del paso de variables, para lo cual, solo hay que agregar un método a la clase:

```
RestfulWebServicesApplication.java HolaMundoController.java *HolaMundoBean.java
1 package com.backfunds.rest.webservices.restfulwebservices.helloworld;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.PathVariable;
5 import org.springframework.web.bind.annotation.RestController;
6
7 @RestController
8 public class HolaMundoController {
9
10    @GetMapping(path = "/hola-mundo")
11    public String holaMundo() {
12        return "Hola Mundo";
13    }
14
15    @GetMapping(path = "/hola-mundo-bean")
16    public HolaMundoBean holaMundoBean() {
17        return new HolaMundoBean("Hola Mundo");
18    }
19
20    @GetMapping(path = "/hola-mundo/path-variable/{nombre}")
21    public HolaMundoBean holaMundoPathVariable(@PathVariable String nombre) {
22        return new HolaMundoBean(String.format("Hola Mundo, %s", nombre));
23    }
24 }
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Y la invocación y salida sería la siguiente:



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

En esta experiencia educativa pudiste aprender a utilizar algunas de las características de Java, desde sus formas más básicas como lo son las variables y los tipos de datos, hasta cómo declarar clases y cómo crear objetos. Reafirmaste algunos conceptos de la Programación Orientada a Objetos y aprendiste a utilizar algunos de los objetos más complejos como lo son los arreglos.

Además, se abordó de manera teórica y práctica el tema de los microservicios y de las API RESTful, es importante que los conocieras, ya que en la actualidad son parte de las soluciones que se utilizan dentro de la industria del software en los tiempos del cómputo distribuido y los microservicios.

Referencias bibliográficas

- IBM. (2021). *Arquitectura J2EE*. Recuperado de <https://www.ibm.com/docs/es/rsm/7.5.0?topic=applications-j2ee-architecture>
- Romero, L. (2021). *Introducción a Spring Boot: Creación de un microservicio – Parte*. Recuperado de <https://ifgeekthen.nttdata.com/es/introduccion-a-spring-boot-creacion-de-un-microservicio>
- Oracle. (s.f.). *Desarrollo de una aplicación Java RESTful basada en microservicios*. Recuperado de <https://docs.oracle.com/es/solutions/develop-microservice-java-app/index.html#GUID-7B47D70F-9DD2-46FC-8E3D-5C6F8C775E54>
- RedHat. (2020). *¿Qué es una API REST?* Recuperado de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>

Para saber más

Lecturas

Para conocer más acerca de **programación en Java**, te sugerimos leer lo siguiente:

- RedHat. (2023). *¿Qué son y para qué sirven los microservicios?* Recuperado de <https://www.redhat.com/es/topics/microservices>
- Java. (s.f.). *¿Qué es la tecnología Java y por qué la necesito?* Recuperado de https://www.java.com/es/download/help/whatis_java.html

Videos

Para conocer más acerca de **programación en Java**, te sugerimos revisar lo siguiente:

- Lucas Moy. (2021, 2 de junio). *Curso completo de Java para principiantes* (2021) [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=Z8zAKYLZBqc>
- Wanna Code. (2021, 29 de noviembre). *Microservicios con Spring Boot – Creando el primer Microservicio* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=-ksmE3KoX9U>

Checkpoints

Asegúrate de:

- Comprender la importancia de Java para su uso en las empresas.
- Entender el funcionamiento de la programación orientada a objetos implementada en Java.
- Comprender los conceptos de los servicios web Rest y sus características.

Requerimientos Técnicos

- JDK de Java.
- Editor de código.

Prework

- Descarga el JDK de Java desde el sitio oficial de acuerdo con tu sistema operativo:



Oracle. (s.f.). *Java SE 16 Archive Downloads.*
Recuperado de
<https://www.oracle.com/java/technologies/javase/jdk16-archive-downloads.html>

El uso y descarga del software deberá apegarse a los términos y condiciones del sitio oficial del fabricante y su uso será responsabilidad de quien lo descargue. Tecmilenio no tiene licencia ni posee los derechos sobre dicho software.

- Para desarrollar, es necesario que tengas instalado un IDE, en este caso se sugiere utilizar Visual Studio Code (VSC), puedes descargarlo del sitio oficial:



Visual Studio Code. (s.f.). *Download Visual Studio Code.*
Recuperado de <https://code.visualstudio.com/download>

El uso y descarga del software deberá apegarse a los términos y condiciones del sitio oficial del fabricante y su uso será responsabilidad de quien lo descargue. Tecmilenio no tiene licencia ni posee los derechos sobre dicho software.

- Una vez que tengas instalado el VSC, instala la extensión “Extension Pack for Java” desarrollado por Microsoft.
- Cuando estés listo, abre la consola en VSC y teclea el comando: java –version y te debe mostrar la versión instalada del JDK.

```
java version "16.0.2" 2021-07-20
Java(TM) SE Runtime Environment (build 16.0.2+7-67)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.2+7-67, mixed mode, sharing)
```

- Si todo salió bien, estás listo para comenzar con tus prácticas.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.