

Java

TechGuide - Alura, FIAP e PM3

Nivel 1

Java - Fundamentos:

- Java es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web. Java es un lenguaje multiplataforma, orientado a objetos y centrado en red que se puede utilizar como una plataforma en sí. Es un lenguaje de programación rápido, seguro y confiable para codificar todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de big data y tecnologías de servidor.
- Conocer los tipos primitivos.
- Declarar variables, considerando los diferentes tipos.
- Usar estructuras condicionales ('if', 'Else').
- Conocer a los operadores de comparación.
- Utilizar estructuras de repetición y bucles ('while', 'for').
- Usar funciones, pasar parámetros y argumentos.
- Manipular métodos.
- Manipular arrays y listas.
- Obtener datos de una API.
- Hacer llamadas asíncronas 'Future', etc.
- Crear constructores.

Contenidos

- **Podcast** Qué se necesita para ser programador junior en java
(<https://open.spotify.com/episode/0sBiBS6TpdHjRUzm1oQm2O?si=b2a759c65ca44dd5>)
- **Artículo** ¿Qué es la tecnología Java y por qué la necesito?
(https://www.java.com/es/download/help/whatis_java.html)

Contenidos Alura:

- **Artículo** Java: una guía para iniciar en esta tecnología (<https://www.aluracursos.com/blog/java-una-guia-para-iniciar-tecnologia>)
- **YouTube** ¿Por qué utilizar Java actualmente? (<https://www.youtube.com/watch?v=3kNuK-XAHEY&t=1s>)
- **YouTube** La magia detrás de Java (<https://www.youtube.com/watch?v=GrEO8nZzyZM&t=34s>)
- **Artículo** ¿Cómo empezar a desarrollar en java? (<https://www.aluracursos.com/blog/como-empezar-a-desarrollar-en-java>)
- **Artículo** Las características más destacables de Java 8 en adelante (<https://www.aluracursos.com/blog/caracteristica-destacables-java8-delante>)
- **Artículo** Java: Conozca el método main (<https://www.aluracursos.com/blog/java-conozca-el-metodo-main>)
- **Artículo** Importando clases en Java (<https://www.aluracursos.com/blog/importando-clases-en-java>)
- **Artículo** Como hacer un import static en java (<https://www.aluracursos.com/blog/como-hacer-un-import-static-en-java>)
- **Artículo** Recibiendo datos en Java (<https://www.aluracursos.com/blog/recibiendo-datos-en-java>)
- **Curso** Java: creando tu primera aplicación (https://app.aluracursos.com/course/java-creando-primer-a-aplicacion?utm_source=gnarus&utm_medium=timeline)

Conceptos de Orientación a Objetos:

- La Programación Orientada a Objetos es un paradigma de programación de software basado en la composición e interacción entre diversas unidades llamadas 'objetos' y las clases, que contienen una identidad, propiedades y métodos. Se basa en cuatro componentes de la programación - abstracción digital, encapsulación, herencia y polimorfismo.
- Cómo funcionan los objetos
- Crear y utilizar constructores
- Qué son las clases
- Crear y utilizar métodos
- Cómo funciona la encapsulación
- Qué es la herencia
- Qué es el polimorfismo

- Cómo funcionan las interfaces
- Qué son las abstracciones

Contenidos

- **Web** Programación orientada a objetos (C#) (<https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/tutorials/oop>)
- **Web** Programación orientada a objetos - IBM (<https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming>)
- **Web** Introducción a los objetos JavaScript (<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects>)
- **Artículo** Introducción a POO con JavaScript ES6 (<https://medium.com/academia-hack/introducci%C3%B3n-a-poo-con-javascript-es6-80074fde0cdf>)
- **Artículo** Programación Orientada a Objetos (<https://ellibrodepython.com/programacion-orientada-a-objetos-python>)
- **YouTube** 4 Principios de la Programación Orientada a Objetos (<https://www.youtube.com/watch?v=UK1C1NARZjU>)
- **YouTube** Mini Curso: POO con PHP (básico) (https://youtu.be/Ben_VC2rm10)
- **YouTube** 🟡 ¿Qué es la Programación Orientada a Objetos en PYTHON? - [Con EJEMPLOS] | Python desde CERO #13 (<https://www.youtube.com/watch?v=KwT1F7uL5rA>)

Contenidos Alura:

- **YouTube** Alura Latam: ¿Qué es la Programación Orientada a Objetos? (<https://www.youtube.com/watch?v=Oigen2sjagk>)
- **Artículo** POO: ¿Qué es la programación orientada a objetos? (<https://www.aluracursos.com/blog/poo-que-es-la-programacion-orientada-a-objetos>)
- **Artículo** Revisando la Orientación a Objetos: encapsulación de Java (<https://www.aluracursos.com/blog/revisando-la-orientacion-a-objetos-encapsulacion-de-java>)
- **Artículo** ¿Qué es encapsulamiento? (<https://www.aluracursos.com/blog/Que-es-encapsulamiento>)
- **Artículo** Herencia en JavaScript (<https://www.aluracursos.com/blog/herencia-en-javascript>)
- **Artículo** Interfaces Gráficas con Eclipse WindowBuilder (<https://www.aluracursos.com/blog/interfaces-graficas-con-eclipse-windowbuilder>)
- **Artículo** Cómo no aprender Java y Orientación a Objetos: getters y setters (<https://www.aluracursos.com/blog/como-no-aprender-java-y-orientacion-a-objetos>)

[getters-y-setters\)](#)

- **Artículo** Ordenar una lista de objetos en Java (<https://www.aluracursos.com/blog/ordenar-una-lista-de-objetos-en-java>)
- **Curso** Formación Java Orientado a Objetos (<https://app.aluracursos.com/formacion-javaoo>)
- **Curso** JavaScript: Introducción a la Orientación a Objetos (<https://app.aluracursos.com/course/javascript-introduccion-orientacion-objetos>)
- **Curso** JavaScript: Herencia e Interfaces en Orientación a Objetos (<https://app.aluracursos.com/course/javascript-herencia-interfaces-orientacion-objetos>)
- **Curso** C# y Orientación a Objetos (<https://app.aluracursos.com/formacion-c-sharpe-orientacion-a-objetos>)
- **Curso** Python: comprensión de la Orientación a Objetos (<https://app.aluracursos.com/course/python-comprension-orientacion-objetos>)
- **Curso** Python: avanzando en la orientación a objetos (<https://app.aluracursos.com/course/python-avanzando-orientacion-objetos>)
- **Curso** Java Polimorfismo: Entendiendo herencia e interfaces (<https://app.aluracursos.com/course/java-parte-3-entendiendo-herencia-interfaces>)
- **Curso** Curso de JavaScript: Objetos (https://app.aluracursos.com/course/javascript-objetos?utm_source=gnarus&utm_medium=timeline)
- **Curso** Formación: Aprenda a programar en Python con orientación a objetos (<https://app.aluracursos.com/formacion-lenguaje-python>)

Java - Manejo de Errores:

- El tratamiento de errores se refiere a los procedimientos de respuesta y recuperación de condiciones de error presentes en una aplicación de software. En otras palabras, es el proceso compuesto de anticipación, detección y resolución de errores de aplicación, de programación o de comunicación.
- Tratamiento de excepciones predefinidas
- Uso de 'Try' y 'catch'
- Crear excepciones específicas
- Hacer el proceso de depuración

Contenidos

- **Web** Breve introducción a las excepciones en Java (https://www.it.uc3m.es/java/prog/resources/excepciones/index_es.html)
- **Web** Excepciones (<https://catedu.github.io/poo-java/excepciones.html>)

- **Web** Excepción - Java Documentation (en inglés) (<https://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html>)
- **Web** Excepciones - Oracle (en inglés) (<https://dev.java/learn/exceptions/>)
- **Web** Excepciones Java - Documentación de IBM (<https://www.ibm.com/docs/es/i/7.1?topic=driver-java-exceptions>)
- **Artículo** Uso de excepciones en Java | IfgeekthenNTTdata (<https://ifgeekthen.nttdata.com/es/uso-de-excepciones-en-java>)
- **Artículo** Tipos de excepciones en Java (<https://www.linkedin.com/pulse/tipos-de-excepciones-en-java-oscar-eduardo-romero-mel%C3%A9ndez/?originalSubdomain=es>)
- **YouTube** THROW 🌈 en JAVA ☕ Lanzar EXCEPCIONES ⚡ - YouTube (<https://www.youtube.com/watch?v=B6h6plcj0WA>)
- **YouTube** 🚀 EXCEPCIONES en JAVA POO 🖥️ | TRY y CATCH | Explicación FÁCIL ⚡ #17 (https://www.youtube.com/watch?v=VHy6xFXJ1Rw&list=TLPQMDcwMzlwMjNzINQIG4Bhnw&index=4&ab_channel=TodoCode)

Contenidos Alura:

- **Curso** Java Excepciones: Aprenda a crear, lanzar y controlar excepciones (<https://app.aluracursos.com/course/java-excepciones>)

Java - Colecciones:

- Una colección representa un grupo de objetos, conocidos como sus elementos. Son como contenedores que agrupan varios elementos en una sola unidad. Algunas colecciones permiten la duplicación de elementos y otras no. Algunas son ordenadas y otras no ordenadas.
- Aprender los usos y diferencias entre List, Set y Map
- Aprender los usos y diferencias entre Equals y hashCode
- Aprende a trabajar con ArrayList, LinkedList o Vector

Contenidos

- **Web** Colección - Java Documentation (en inglés) (<https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html>)
- **Artículo** Introducción a Colecciones en Java (<https://www.adictosaltrabajo.com/2015/09/25/introduccion-a-colecciones-en-java/>)
- **Artículo** Map en Java, con ejemplos (<https://jarroba.com/map-en-java-con-ejemplos/>)
- **YouTube** 🚀 ¿Qué son las COLLECTIONS en JAVA? POO 🖥️ | LIST, SET, QUEUE, MAP | Explicación FÁCIL ⚡ #12 (https://www.youtube.com/watch?v=mE1vnyn_QgU&ab_channel=TodoCode)

- **YouTube** HASHCODE y EQUALS en JAVA - Tutorial Completo Fácil - YouTube (<https://www.youtube.com/watch?v=4qjcWlnm6dg>)

Contenidos Alura:

- **Curso** Curso Java y java.util: Colecciones, Wrappers y Lambda expressions (<https://app.aluracursos.com/course/java-util-colecciones-wrappers-lambda-expressions>)

Java - Pruebas:

- La prueba de software es el proceso de evaluación y verificación de que un software realmente hace lo que debe hacer. Los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento.
- Usar pruebas unitarias
- Usar pruebas de integración
- Usar pruebas de comportamiento (behavior)
- Usar mocks

Contenidos

- **Artículo** Tutorial de JUnit: Configuración, escritura y ejecución de pruebas unitarias de Java (<https://es.parasoft.com/blog/junit-tutorial-setting-up-writing-and-running-java-unit-tests/>)
- **Artículo** Burlarse en Java: cómo automatizar una prueba unitaria de Java, incluidas las burlas y las afirmaciones (<https://es.parasoft.com/blog/mocking-in-java-how-to-automate-a-java-unit-test-including-mocking-and-assertions/>)
- **Artículo** Testeando aplicaciones en Java (<https://medium.com/@yonem9/testeando-aplicaciones-en-java-c46364191562>)
- **YouTube** PRUEBAS UNITARIAS en JAVA (JUNIT 5) (https://www.youtube.com/watch?v=74sCIDEYSQ4&ab_channel=ProgramandoenJAVA)
- **YouTube** MOCK vs STUB Aquí encuentras las diferencias ✅ (https://www.youtube.com/watch?v=ArkckkM49LQ&ab_channel=CodingTogetherES)
- **Artículo** JUnit: Cómo ejecutar un test N iteraciones (<https://dev.to/gekyzo/como-repetir-un-test-n-iteraciones-36pd>)
- **Artículo** JUnit Cómo parametrizar un test (<https://dev.to/gekyzo/junit-como-ejecutar-un-test-con-parametros-4cp3>)
- **Artículo** JUnit Cómo repetir un test con parámetros (<https://dev.to/gekyzo/junit-como-repetir-un-test-con-parametros-1l15>)

Contenidos Alura:

- **Artículo** Tipos de pruebas: ¿cuáles son las principales y por qué utilizarlas? (<https://www.aluracursos.com/blog/tipos-de-pruebas-cual-utilizar>)
- **YouTube** ¿Qué es Test-Driven Development (TDD)? #AluraMás (https://www.youtube.com/watch?v=qA3Klo0xZbE&t=1s&ab_channel=AluraLatam)
- **Curso** Curso Java y Tests: Test Driven Development con Junit (<https://app.aluracursos.com/course/java-tests-test-driven-development-junit>)

Java - Paquetes:

- Un paquete (package) en Java se utiliza para agrupar clases relacionadas, de forma similar a una carpeta en un directorio de archivos. Los paquetes se utilizan para evitar conflictos de nombres y para escribir un código de mejor mantenimiento.
- Utilice Imports y organice su código a través de paquetes (packages)
- Conocer la java.lang
- Comprender la inmutabilidad y la clase String
- Entender la clase java.lang.Object
- Conocer la java.io

Contenidos

- **Web** Java.Lang Espacio de nombres (<https://learn.microsoft.com/es-es/dotnet/api/java.lang?view=xamarin-android-sdk-13>)
- **Artículo** Java - Paquetes (<https://www.arkaitzgarro.com/java/capitulo-19.html>)
- **YouTube** Uso de Paquetes en java (<https://www.youtube.com/watch?v=-cj9H8qLAcU>)
- **YouTube** Crear y utilizar nuestros propios paquetes de Java (<https://www.youtube.com/watch?v=y2OZYSSoysY>)

Contenidos Alura:

- **Curso** Java y java.lang: Programe con la clase Object y String (<https://app.aluracursos.com/course/java-lang-clase-object-string>)

Estructura de Datos:

- En el contexto de los ordenadores, una estructura de datos es una forma específica de almacenar y organizar los datos en la memoria del ordenador para que esos datos puedan ser fácilmente recuperados y utilizados de forma eficiente cuando sea necesario posteriormente.
- Conocer las principales estructuras de datos

- Implementar las principales estructuras de datos

Contenidos

- **Artículo** ¿Qué es una estructura de datos en programación y para qué se utiliza? (<https://blog.soyhenry.com/que-es-una-estructura-de-datos-en-programacion/>)
- **Web** Estructuras de datos - Documentación Python (<https://docs.python.org/es/3/tutorial/datastructures.html>)
- **Artículo** Estructuras de Datos y Algoritmos en Java (https://programacion.net/articulo/estructuras_de_datos_y_algoritmos_en_java_309/4)
- **Artículo** Estructuras de datos con Java: un enfoque práctico (<http://hp.fciencias.unam.mx/~alg/estructurasDeDatos/>)
- **Artículo** Colecciones: ARRAY, SET y DICTIONARY en Swift (<https://www.swiftbeta.com/colecciones-array-set-y-dictionary-en-swift/>)
- **Artículo** Colecciones en Swift y su manejo de memoria (<https://medium.com/@grago/colecciones-en-swift-y-su-manejo-de-memoria-61a03e236dd>)
- **Artículo** Estructuras de datos en .NET con C# (<https://www.genesisrrios.com/es/blog/estructuras-de-datos-en-csharp/>)
- **YouTube** Estructuras de Datos | Primeros Pasos (<https://youtu.be/Df-sgxGzyTg>)
- **YouTube** Qué son las estructuras de datos (<https://youtu.be/oQ0WklDr73E?list=PLTd5ehlj0goMTSK7RRAPBF4wP-Nj5DRvT>)
- **YouTube** ¿Qué son y cómo funcionan los árboles? | Ejemplo de implementación (<https://youtu.be/tBaOQeyXYqg>)
- **YouTube** Aprende: Estructura de Datos con Java (https://www.youtube.com/watch?v=_9ScDWpqhFE)
- **YouTube** Introducción - 1 - Estructuras de Datos en C# (https://www.youtube.com/watch?v=rqaqJXbauyA&ab_channel=nicosiored)
- **YouTube** Estructuras de datos con Python en 8 minutos: Listas, Tuplas, Conjuntos y Diccionarios (<https://www.youtube.com/watch?v=v25-m1LOUiU>)
- **YouTube** Programación en Python | Colecciones | Diccionarios (<https://www.youtube.com/watch?v=vAy4IM7NLIQ>)

Contenidos Alura:

- **Artículo** Estructura de datos: introducción (<https://www.aluracursos.com/blog/estructura-de-datos-introduccion>)

- **Artículo** Conociendo tuplas en Python (<https://www.aluracursos.com/blog/conociendo-las-tuplas-en-python>)
- **Artículo** Listas en Python: operaciones básicas (<https://www.aluracursos.com/blog/listas-de-python-operaciones-basicas>)
- **Artículo** Estructura de datos: computación práctica con Java (<https://www.aluracursos.com/blog/estructura-de-datos-computacion-practica-con-java>)
- **Artículo** Python: trabajando con diccionarios (<https://www.aluracursos.com/blog/python-trabajando-con-diccionarios>)
- **Artículo** Iterando una lista en Java (<https://www.aluracursos.com/blog/iterando-una-lista-en-java>)
- **Curso** Python: avanzando en el lenguaje (<https://app.aluracursos.com/course/python-avanzando-lenguaje>)
- **Curso** C#: array y tipos genéricos (<https://app.aluracursos.com/course/csharp-array-tipos-genericos>)
- **Curso** Python Collections: listas y tuplas (https://app.aluracursos.com/course/python-collections-listas-tuplas?utm_source=gnarus&utm_medium=timeline)
- **Curso** Curso de Java: trabajar con listas y colecciones de datos (<https://www.aluracursos.com/curso-online-java-trabajar-listas-colecciones-datos>)
- **Curso** Java y java.util: Colecciones, Wrappers y Lambda expressions (<https://www.aluracursos.com/curso-online-java-util-colecciones-wrappers-lambda-expressions>)

Nivel 2

JVM:

- Máquina virtual Java (en inglés, Java Virtual Machine, JVM) es un programa que carga y ejecuta las aplicaciones Java, convirtiendo los bytecodes en código ejecutable de máquina. JVM es responsable de administrar las aplicaciones a medida que se ejecutan. Gracias a la máquina virtual Java, los programas escritos en Java pueden funcionar en cualquier plataforma de hardware y software que tenga una versión de JVM, haciendo así que estas aplicaciones sean independientes de la plataforma donde funcionan.
- Entender cómo funciona la máquina virtual de Java.

Contenidos

- **Artículo** Tutorial JVM - La arquitectura de la máquina virtual de Java explicada para principiantes (<https://www.freecodecamp.org/espanol/news/tutorial-jvm-la-arquitectura->

[de-la-maquina-virtual-de-java-explicada-para-principiantes/](#))

- **YouTube** ¿Cómo Funciona la Java Virtual Machine? (<https://www.youtube.com/watch?v=IEzqeKBOHIY>)

Contenidos Alura:

- **YouTube** La magia detrás de Java #aluramás (<https://www.youtube.com/watch?v=GrEO8nZzyZM&t=1s>)
- **Artículo** Paso a paso con la Máquina Virtual de Java (JVM) (<https://www.aluracursos.com/blog/paso-a-paso-jvm>)

Java - Administración de Memoria:

- En Java, la administración de memoria es el proceso de asignación y desalojo de objetos, llamado administración de memoria. Java administra la memoria automáticamente. Java utiliza un sistema de gestión automática de memoria llamado Garbage Collector (colector de basura). Por lo tanto, no estamos obligados a implementar la lógica de gestión de memoria en nuestra aplicación.
- Entender cómo funciona la memoria y su gestión en Java.
- Entender cómo funciona la memoria de Garbage Collector.

Contenidos

- **Artículo** Administración de memoria Java (<https://learn.microsoft.com/es-es/azure/spring-apps/concepts-for-java-memory-management>)
- **YouTube** ¿Cómo funciona la memoria en Java? El uso de Stack y Heap (https://www.youtube.com/watch?v=zSRO_b4y2cA)
- **YouTube** Tutorial Java - como programar - 12 Objetos: Montón y Recolector de Basura (<https://www.youtube.com/watch?v=LHqWyVFA1l8>)

Spring Framework:

- Spring es un framework de código abierto para la plataforma Java. Se trata de un marco no intrusivo, basado en los estándares de diseño (design patterns) de inversión de control (ioc) e inyección de dependencia. En Spring el contenedor se encarga de "instanciar" clases de una aplicación Java y definir las dependencias entre ellas a través de un archivo de configuración en formato XML, inferencias del framework, lo que es llamado de auto-wiring o incluso anotaciones en las clases, métodos y propiedades. De esta forma, Spring permite el bajo acoplamiento entre clases de una aplicación orientada a objetos.
- Entender el concepto de inyección de dependencias

- Entender el patrón MVC
- Usar Spring Data para manipular datos

Contenidos

- **Artículo** ¿Qué son Spring framework y Spring Boot? Tu primer programa Java con este framework (https://dev.to/campusmvp_es/que-son-spring-framework-y-spring-boot-tu-primer-programa-java-con-este-framework-4ao6)
- **Artículo** Que es la inyección de dependencias en Spring (<https://gustavopeiretti.com/spring-inyeccion-dependencias/>)
- **YouTube** Aprende Spring Boot de una vez por todas! #17 Inyección de Dependencias (https://www.youtube.com/watch?v=SEaDMUy0xKo&ab_channel=CodigoMorsa)
- **YouTube** Aprende Spring Boot de una vez por todas! #19 Beans (https://www.youtube.com/watch?v=ZCy8JzmZi8Q&ab_channel=CodigoMorsa)
- **Artículo** Que es el Modelo Vista Controlador (MVC) (<https://peznuss.medium.com/que-es-el-modelo-vista-controlador-mvc-5b7ff4dbf85e>)
- **Artículo** Conversores personalizados para @RequestBody (<https://dev.to/gekyzo/conversores-personalizados-para-requestbody-37hp>)
- **Artículo** Tips para usar Spring JdbcTemplate (<https://medium.com/@matedeilo/tips-para-usar-spring-jdbctemplate-486d250dc5a>)
- **YouTube** JPARepository en SPRING DATA - Tutorial Completo Fácil (https://www.youtube.com/watch?v=6VyOrS9Obog&ab_channel=ProgramandoenJAVA)

Contenidos Alura:

- **YouTube** Spring Framework. ¿Qué es? #AluraMás (<https://www.youtube.com/watch?v=t-igt1b2qqk&t=60s>)
- **Artículo** Spring: Conozca este framework de Java (<https://www.aluracursos.com/blog/spring-conozca-framework-java>)
- **Artículo** Empezando con Spring Framework (<https://www.aluracursos.com/blog/empezando-con-spring-framework>)
- **Curso** Curso de Spring MVC: crea un web app con Thymeleaf y Bootstrap (<https://app.aluracursos.com/course/spring-mvc-crea-web-app-thymeleaf-bootstrap>)
- **Artículo** Programando tareas con Scheduled de Spring (<https://www.aluracursos.com/blog/programando-tareas-con-scheduled-de-spring>)

Spring Boot:

- Spring Boot es un framework de código abierto basado en Java que se utiliza para crear Microservicios con Spring Framework. Se utiliza para construir aplicaciones de Spring

independientes y listas para la producción.

- Crear aplicaciones independientes de Spring
- Usar los servidores HTTP integrados

Contenidos

- **Web** Spring Boot - Documentación (en inglés pero puede traducir por navegador) (<https://spring.io/projects/spring-boot>)
- **YouTube** ¿Qué es Spring Boot y por qué deberías usarlo? (https://www.youtube.com/watch?v=wIZWt_flAyM)
- **YouTube** Tutorial Spring boot sencillo y en español (<https://www.youtube.com/watch?v=WVHnk04skPc>)
- **YouTube** Spring Boot + Java: Junior Developer lo que deberías saber para el primer trabajo (<https://www.youtube.com/watch?v=q7WA7y-9Fjg>)
- **YouTube** Autowired en Spring Boot - Tutorial Completo Fácil (https://www.youtube.com/watch?v=D1KpLFCuhcY&ab_channel=ProgramandoenJAVA)

Contenidos Alura:

- **Curso** Curso de Spring Boot 3: desarrollar una API Rest en Java (<https://app.aluracursos.com/course/spring-boot-3-desarrollar-api-rest-java>)
- **Curso** Curso de Spring Boot: Seguridad de la API, caché y monitoreo (<https://app.aluracursos.com/course/spring-boot-seguridad-api-cache-monitoreo>)

Builds Tools Java:

- Un build tool es un sistema que permite automatizar todas las tareas rutinarias de un proyecto de una manera organizada y que evite que el desarrollador tenga que perder tiempo. En otras palabras, agregar una nueva biblioteca, realizar pruebas, empaquetar y desplegar, o incluso, la compatibilidad entre varios IDEs son tareas fácilmente resueltas con una herramienta de construcción.
- Descubre las principales herramientas de construcción de Java, como Maven, Jenkins, Apache Ant, Gradle, etc., y cómo usarlas.

Contenidos

- **Artículo** ¿Qué es un Build? (<https://www.itdo.com/blog/que-es-un-build/>)
- **Web** Apache Maven Project (en inglés pero puede traducir por navegador) (<https://maven.apache.org/>)
- **Web** Gradle Build Tool (en inglés pero puede traducir por navegador) (<https://gradle.org/>)

- **YouTube** ★ ¿Qué es MAVEN? ¿Cómo usar MAVEN? MAVEN vs Gradle 🖥️ | DESARROLLO JAVA 🌐 (<https://www.youtube.com/watch?v=rE7zLuQv2IU>)
- **YouTube** 🤖 Crear proyecto java web con 🖱️ MAVEN 🖱️ (<https://www.youtube.com/watch?v=UOWC2hmv2Us>)
- **YouTube** Aprende Gradle desde cero (<https://www.youtube.com/watch?v=53dmgTdWLnk>)
- **Web** Construir y probar Java con Gradle (<https://docs.github.com/es/enterprise-cloud@latest/actions/automating-builds-and-tests/building-and-testing-java-with-gradle>)
- **Web** Construir y probar Java con Ant (<https://docs.github.com/es/enterprise-cloud@latest/actions/automating-builds-and-tests/building-and-testing-java-with-ant>)

Contenidos Alura:

- **Artículo** ¿Qué es Maven? (<https://www.aluracursos.com/blog/que-es-maven>)

Java - Persistencia:

- El concepto de "persistencia de datos" se refiere a garantizar que la información introducida en la aplicación se almacena en un medio en el que se puede recuperar de forma consistente. Es decir, son registros permanentes y que no se pierden cuando se cierra la sesión.
- Entender sobre JDBC y JPA
- Usar frameworks como Spring Data e Hibernate
- Comunicarse con una base de datos relacional
- Entender la diferencia entre las relaciones EAGER y LAZY
- Planificar consultas con Join fetch
- Encapsular el acceso en un DAO (Objeto de Acceso a Datos, en inglés Data Access Objeto)
- Entender cómo funciona la memoria en esta situación

Contenidos

- **Artículo** Persistencia Básica en Java (<https://learn.microsoft.com/es-es/azure/spring-apps/concepts-for-java-memory-management>)
- **YouTube** Ejemplo Java + JDBC – 2. Crear los DAO (<https://www.youtube.com/watch?v=NjY-WA-jeJ8>)

Contenidos Alura:

- **Artículo** Conociendo JDBC (<https://www.aluracursos.com/blog/conociendo-el-jdbc>)

- **Artículo** Bases de datos relacionales (<https://www.aluracursos.com/blog/base-de-datos-relacional>)
- **Artículo** Normalización en base de datos - Estructura (<https://www.aluracursos.com/blog/normalizacion-en-base-de-datos>)
- **Artículo** Entendiendo el Lazy y el Eager Load de JPA (<https://www.aluracursos.com/blog/entendiendo-el-lazy-y-el-eager-load-de-jpa>)
- **YouTube** Descomplicando Base de Datos (<https://www.youtube.com/watch?v=G1cDRqKuxpg>)
- **Curso** Persistencia con JPA: Hibernate (<https://app.aluracursos.com/course/persistencia-jpa-hibernate>)
- **Curso** Java y JDBC: Trabajando con una Base de Datos (<https://app.aluracursos.com/course/java-jdbc-trabajando-base-datos>)

Nivel 3

Arquitectura de Microservicios:

- Los microservicios son un enfoque de arquitectura en el que el software consiste en pequeños servicios independientes que se comunican entre sí y se organizan de acuerdo con sus dominios de negocio.
- Aprender el concepto de arquitectura diseñada para microservicios
- Realizar la comunicación mediante API
- Mejorar la escalabilidad de un sistema

Contenidos

- **Web** Diseño de una aplicación orientada a microservicios (<https://learn.microsoft.com/es-es/dotnet/architecture/microservices/multi-container-microservice-net-applications/microservice-application-design>)
- **Artículo** Microservicios y arquitectura de microservicios (<https://www.intel.la/content/www/xl/es/cloud-computing/microservices.html>)
- **Artículo** Microservicios vs API: Entendiendo la diferencia (<https://kinsta.com/es/blog/microservicios-vs-api/>)
- **Artículo** Microservicios Ejemplo de Flujo (<https://medium.com/mycodebad/microservicios-ejemplo-de-flujo-f45720a9b278>)
- **Artículo** Microservicios Conceptos (<https://medium.com/mycodebad/microservicios-conceptos-55d19636873e>)

- **YouTube** Patrones fundamentales de la arquitectura microservicios (<https://www.youtube.com/watch?v=A7y23uU1NHk>)
- **YouTube** Un ejemplo de microservicios #CafeConRivas (<https://www.youtube.com/watch?v=qAcUGw7HhxM>)

Java - Concurrencia:

- Programación concurrente es un paradigma de programación para la construcción de programas que hacen uso de la ejecución simultánea de varias tareas computacionales interactivas, que pueden ser implementadas como programas separados o como un conjunto de hilos creados por un único programa.
- Realizar tareas simultáneamente.
- Poner tareas a esperar hasta que ocurra un evento determinado.
- Entender cómo funciona la memoria en esta situación.

Contenidos

- **Artículo** Introducción a la Concurrencia en Java (I) (<https://blog.softtek.com/es/java-concurrency>)
- **Artículo** Programación concurrente (https://ferestrepoca.github.io/paradigmas-de-programacion/progconcurrente/concurrente_teoría/index.html)
- **Artículo** Java - Threads (https://informatica.uv.es/iiguia/LP/teoria/apuntes/cuatr1/tema3_2_concurrencia2.pdf)
- **YouTube** Programando en JAVA: HILOS (THREAD) en JAVA (<https://www.youtube.com/watch?v=tpnasHx9WLg>)
- **YouTube** Concurrencia estructurada en Java, todo lo que necesitas saber (<https://www.youtube.com/watch?v=l1lIZUGhHEY>)



Contenedores:

- Los contenedores son paquetes de software que contienen todos los elementos necesarios para ejecutarse en cualquier entorno. La gestión de contenedores es un área crucial en la computación en nube y DevOps, que implica el uso de tecnologías para automatizar el proceso de creación, implementación, escalado y monitoreo de contenedores. Los contenedores son unidades de software estandarizadas que permiten a los desarrolladores empaquetar todas las dependencias de una aplicación (código, bibliotecas, configuraciones, etc.) en un solo paquete. Esto permite que la aplicación se ejecute de forma consistente en cualquier entorno de infraestructura.
- La tecnología de contenedores, como ejemplifica Docker, proporciona un entorno coherente y portátil para el desarrollo, las pruebas y la implementación de aplicaciones,

lo que es vital para el trabajo eficiente de la ingeniería de datos. Además, Kubernetes, un sistema de organización de contenedores, permite la gestión, automatización y escalabilidad de aplicaciones basadas en contenedores en entornos de producción. Dominar estos conceptos y tecnologías permite a los ingenieros de datos construir y mantener canalizaciones de datos eficientes y confiables.

- Kubernetes (también conocido como k8s o Kube) es una plataforma de orquestación de contenedores de código abierto que automatiza gran parte de los procesos manuales necesarios para implementar, gestionar y escalar aplicaciones en contenedores.
- Aislar el software para que funcione independientemente
- Implementación de software en clústeres
- Modularizar su sistema en paquetes más pequeños
- Conocer la plataforma Docker
- Conocer Kubernetes

Contenidos

- **Web** IBM - ¿Qué son los contenedores? (<https://www.ibm.com/mx-es/topics/containers>)
- **Web** Microsoft - ¿Qué es un contenedor? (<https://azure.microsoft.com/es-mx/resources/cloud-computing-dictionary/what-is-a-container>)
- **Artículo** Contenedores y la nube (<https://dev.to/roremdev/contenedores-y-la-nube-3je9>)
- **Artículo** Contenedores: cómo es el ciclo de vida de una aplicación en Kubernetes (<https://dev.to/campusmvp/contenedores-como-es-el-ciclo-de-vida-de-una-aplicacion-en-kubernetes-1ief>)
- **Artículo** ¿Qué diferencia hay entre Docker (Contenedores) y Máquinas virtuales (VMWare, VirtualBox...)? (<https://dev.to/campusmvp/qu-diferencia-hay-entre-docker-contenedores-y-mquinas-virtuales-vmware-virtualbox-4ji3>)
- **Artículo** El potencial de Kubernetes para las empresas (<https://jlcasal.medium.com/el-potencial-de-kubernetes-para-las-empresas-d36f9eece999>)
- **Artículo** ¿Qué es Docker y para que sirve? Explicación (https://dev.to/prox_sea/que-es-docker-y-para-que-sirve-explicacion-5h2n)
- **YouTube** ¿Que es un contenedor? (<https://www.youtube.com/watch?v=x5zGoICZLnU>)
- **YouTube** ¿Qué es Docker y los contenedores? (<https://www.youtube.com/watch?v=kkfZs0vJFyU>)
- **YouTube** Volúmenes y redes en Docker (https://www.youtube.com/watch?v=DIIdel70dFI&ab_channel=Programaci%C3%B3nEspa%C3%B1ol)
- **YouTube** ¿Qué es KUBERNETES?  Relación con DOCKER y CONTENEDORES  (<https://www.youtube.com/watch?v=V86eTbswdQo>)

Contenidos Alura:

- **Artículo** Empezando con Docker (<https://www.aluracursos.com/blog/empezando-con-docker>)
- **Artículo** Creando volúmenes con Docker (<https://www.aluracursos.com/blog/creando-volumenes-con-docker>)

Java - Kafka:

- Apache Kafka es una plataforma de transmisión de datos distribuida que es capaz de publicar, suscribir, almacenar y procesar flujos de registro en tiempo real. Esta plataforma está diseñada para procesar flujos de datos provenientes de diversas fuentes y entregarlos a varios clientes.
- Utilizar Kafka para la comunicación asíncrona
- Crear microservicios con Kafka
- Crear productores y consumidores
- Entender cómo usar Kafka para paralelismo y ejecución serializada
- Obtener garantías relativas al envío o entrega de los mensajes

Contenidos

- **YouTube** Apache KAFKA: Qué es en 1 minuto (<https://www.youtube.com/watch?v=bxIsNY8Au9I>)
- **Artículo** Qué es Apache Kafka y cómo dar los primeros pasos (<https://profile.es/blog/que-es-apache-kafka-primeros-pasos/>)
- **Podcast** ¿Qué es y cómo funciona Apache Kafka? Podcast Apasionados por la tecnología. (<https://open.spotify.com/episode/52RjBgXyclVSd2FHkkENQV?si=130fac7df95b4ec6>)
- **Web** Introducción a Apache Kafka (<https://oscarfmdc.medium.com/introducci%C3%B3n-a-apache-kafka-aprende-big-data-en-espa%C3%B1ol-ce7281fdf0cc>)
- **YouTube** Apache Kafka - Ejemplo de configuración de intermediario de Apache Kafka (<https://www.ibm.com/docs/es/oala/1.3.7?topic=collection-apache-kafka-broker-configuration-example>)
- **YouTube** Como comunicar microservicios y ejemplo con Apache Kafka y Javascript (https://www.youtube.com/watch?v=rHRSOWa0cqI&ab_channel=NullSafeArchitect)
- **YouTube** Instalación de un cluster de Apache Kafka paso a paso (https://www.youtube.com/watch?v=MA-nxL14fr4&ab_channel=NullSafeArchitect)

Habilidad Auxiliar: Infraestructura

Git y GitHub - Fundamentos:

- Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia.
- GitHub es un servicio de hosting para el desarrollo de software y el control de versiones mediante Git.
- Crear un repositorio
- Clonar un repositorio
- Comprometerse, empujar y tirar hacia y desde el repositorio
- Revertir un commit
- Crear de ramas y Pull requests
- Manejar fusiones y conflictos

Contenidos

- **Web** Git: Libro de Consulta (<https://git-scm.com/book/es/v2>)
- **Web** GitHub Documentación (<https://docs.github.com/es>)
- **Web** Github Pages Documentación (<https://docs.github.com/es/pages/getting-started-with-github-pages/about-github-pages>)
- **Web** W3Schools: Git Tutorial (<https://www.w3schools.com/git/default.asp?remote=github>)
- **Web** Git School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)
- **Web** Dangit, Git!?! (<https://dangitgit.com/es>)
- **Artículo** Git and Github Quickstart Tutorial (<https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca>)
- **YouTube** ¿Qué es Git y cómo funciona? (<https://www.youtube.com/watch?v=jGehuhFhtnE>)
- **YouTube** Git y Github | Guía Práctico de Git y Github Desde Cero (<https://www.youtube.com/watch?v=HiXLkL42tMU>)

Contenidos Alura:

- **Artículo** Git y Github: que son y primeros pasos (<https://www.aluracursos.com/blog/git-y-github-que-son-y-primeros-pasos>)
- **Artículo** Guía sobre cómo instalar Git en diferentes sistemas operativos (<https://www.aluracursos.com/blog/guia-sobre-como-instalar-git-en-diferentes-sistemas-operativos>)









- **Artículo** Iniciando un repositorio con Git (<https://www.aluracursos.com/blog/iniciando-repositorio-con-git>)
- **Artículo** Comenzando con Git: aprendiendo a versionar (<https://www.aluracursos.com/blog/comenzando-con-git>)
- **Artículo** Creando un repositorio remoto en GitHub (<https://www.aluracursos.com/blog/creando-repositorio-remoto-en-github>)
- **Artículo** Clonando un repositorio con Git y GitHub (<https://www.aluracursos.com/blog/clonando-un-repositorio-remoto>)
- **Artículo** Paso a Paso para activar tu proyecto en GitHub Pages. (<https://www.aluracursos.com/blog/github-pages>)
- **Artículo** Cómo escribir un README increíble en tu Github (<https://www.aluracursos.com/blog/como-escribir-un-readme-increible-en-tu-github>)
- **Artículo** Buenas practicas en git: evitando errores (<https://www.aluracursos.com/blog/como-evitar-errores-en-git>)
- **Artículo** GIT: Errores de comandos y directorios (<https://www.aluracursos.com/blog/git-errores-de-comandos-y-directorios>)
- **Artículo** GIT: errores de commits (<https://www.aluracursos.com/blog/git-errores-de-commits>)
- **Artículo** GIT: Errores de fusión (<https://www.aluracursos.com/blog/errores-de-fusion>)
- **Artículo** GIT: Errores con el remoto (<https://www.aluracursos.com/blog/errores-con-el-remoto>)
- **YouTube** Git y GitHub para Principiantes #AluraMás (https://www.youtube.com/watch?v=LmFK6skG7s&ab_channel=AluraLatam)
- **YouTube** Git y GitHub: Herramientas Esenciales para el Control de Versiones y Colaboración en Desarrollo (<https://youtu.be/dw04N616Abw>)
- **YouTube** ¿Puedo subir mi proyecto a Github sin líneas de comando? - Git y Github para principiantes (https://www.youtube.com/watch?v=Yfm16Tlpcwk&ab_channel=AluraLatam)
- **YouTube** Git y GitHub: Herramientas Esenciales para el Control de Versiones y Colaboración en Desarrollo (<https://www.youtube.com/watch?v=dw04N616Abw>)
- **YouTube** ¿Puedo subir mi proyecto a Github sin líneas de comando? - Git y Github para principiantes (<https://www.youtube.com/watch?v=Yfm16Tlpcwk>)
- **Curso** Git y GitHub: repositorio, commit y versiones (<https://app.aluracursos.com/course/git-github-repositorio-commit-versiones>)
- **Curso** Git y Github: estrategias de ramificación, conflictos y Pull Requests (<https://www.aluracursos.com/curso-online-git-github-estrategias-ramificacion->

[conflictos-pull-requests](#))

HTTP - Fundamentos:

- HTTP significa Protocolo de transferencia de hipertexto. La comunicación entre las computadoras cliente y los servidores web se realiza mediante el envío de solicitudes HTTP y la recepción de respuestas HTTP.
- Comprender la diferencia entre los verbos HTTP
- Probar solicitudes y verificar los códigos de estado en el navegador
- Aprendiendo a hacer una solicitud HTTP en la línea de comando con WGET
- Descargar una imagen con WGET
- Realización de una POST

Contenidos

-  W3Schools: ¿Qué es HTTP? (https://www.w3schools.com/whatis/whatis_http.asp)
-  MDN Web Docs: Una descripción general de HTTP (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>)
-  MDN Web Docs: Métodos de solicitud HTTP (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>)
-  MDN Web Docs: HTTP (<https://developer.mozilla.org/es/docs/Web/HTTP>)
-  MDN Web Docs: Métodos de petición HTTP (<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>)
-  MDN Web Docs: Mensajes HTTP (<https://developer.mozilla.org/es/docs/Web/HTTP/Messages>)
-  HTTP Cats (<https://http.cat/>)
-  HTTP Dogs (<https://http.dog/>)
-  Peticiones, Métodos Http y Códigos de estado. (<https://www.youtube.com/watch?v=gBK-Mfa0lw8>)
-  SSL, TLS, HTTPS, HTTP - Explicado Fácilmente (https://www.youtube.com/watch?v=6HJAWFenYx8&ab_channel=ProfeSang)
-  ⭐ Protocolo HTTP 🖥️ Requests y Responses con: GET, POST, PUT, PATCH y DELETE | Desarrollo web 🌐 (<https://www.youtube.com/watch?v=l2MihYAj0lw>)
-  REST y los Verbos de HTTP (<https://www.youtube.com/watch?v=OHBHeAPoZ8E>)

Contenidos Alura:

- **Artículo** HTTP: Desmitificando el protocolo Web (<https://www.aluracursos.com/blog/http-desmitificando-el-protocolo>)
- **Artículo** ¿Cual es la diferencia entre HTTP y HTTPS? (<https://www.aluracursos.com/blog/cual-es-la-diferencia-entre-http-y-https>)
- **Artículo** HTTP: Diferencias entre GET y POST (<https://www.aluracursos.com/blog/diferencias-entre-get-y-post>)
- **Artículo** Métodos de petición HTTP (<https://www.aluracursos.com/blog/metodos-de-peticion-http>)
- **Curso** HTTP: La base de internet (<https://app.aluracursos.com/course/http-base-internet>)

JSON:

- JSON significa Notación de objetos de JavaScript. Es un formato de texto para almacenar y transportar datos.
- Crear un objeto
- Transformar un objeto en una cadena
- Transformar una cadena en un objeto
- Manipular un objeto

Contenidos

- **Web** MDN Web Docs: JSON (https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/JSON)
- **Web** MDN Web Docs: Trabajando con JSON (<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>)
- **YouTube** ¿Que és JSON y cómo funciona? (https://www.youtube.com/watch?v=z8qk7T_2sWg&ab_channel=SoyDalto)
- **YouTube** ★ ¿Qué es JSON? ¿Cuál es su SINTAXIS? 🖥️ ¿Cómo crear un archivo JSON? | DESARROLLO WEB 🌐 (https://www.youtube.com/watch?v=RhxOTqFbI5Q&ab_channel=TodoCode)

Contenidos Alura:

- **Artículo** ¿Que es Json? (<https://www.aluracursos.com/blog/que-es-json>)
- **Artículo** ¿JSON y Objeto JavaScript son lo mismo? (<https://www.aluracursos.com/blog/json-y-objeto-javascript-son-lo-mismo>)

- **Artículo** Simulando una API REST con json-server
(<https://www.aluracursos.com/blog/simulando-una-api-rest-con-json-server>)
- **Artículo** ¿Qué es JSON Web Token? (<https://www.aluracursos.com/blog/que-es-json-web-token>)
- **Curso** JS en la Web: CRUD con JavaScript asíncrono
(<https://www.aluracursos.com/curso-online-js-web-crud-javascript-asincrono>)

Línea de Comando - Fundamentos:

- CLI es un programa de línea de comandos que acepta la entrada de texto para ejecutar funciones del sistema operativo.
- Conocer los comandos más importantes

Contenidos

- **Web** W3Schools: What is Command Line Interface (CLI)?
(https://www.w3schools.com/whatis/whatis_cli.asp)
- **Web** Uso de argumentos de la línea de comandos para Terminal Windows
(<https://learn.microsoft.com/es-es/windows/terminal/command-line-arguments?tabs=windows>)
- **Artículo** Interfaz de línea de comandos o CLI
(<https://www.computerweekly.com/es/definicion/Interfaz-de-linea-de-comandos-o-CLI>)
- **Artículo** El Manual de Comandos de Linux
(<https://www.freecodecamp.org/espanol/news/comandos-de-linux/>)
- **YouTube** Consola vs Terminal vs Shell vs CLI 🖥️ ¿Qué es la terminal?
(https://www.youtube.com/watch?v=1YxHXBsVNGQ&ab_channel=ProgramadorX)
- **YouTube** Aprende la linea de comandos en un mac - bash scripting
(https://www.youtube.com/watch?v=vfwA3pUnVOg&ab_channel=Datademia)
- **YouTube** freeCodeCamp.org: Command Line Crash Course
(<https://www.youtube.com/watch?v=yz7nYInXLfE>)
- **YouTube** Traversy Media: Command Line Crash Course For Beginners - Terminal Commands
(<https://www.youtube.com/watch?v=uwAqEzhyjtw>)
- **YouTube** Comandos Básicos e Intermedios CMD (<https://youtu.be/erKosEQaaFc>)
- **YouTube** Terminal MAC tutorial en Español - Cómo usar la terminal en MAC
(<https://www.youtube.com/watch?v=TmP3y7Z2kk4>)

Contenidos Alura:

- **Artículo** CMD: Sugerencias para trabajar en el prompt de Windows
(<https://www.aluracursos.com/blog/consejos-para-trabajar-en-el-indicador-de->

[windows](#))

- **Artículo** Como usar el terminal integrado de Visual Studio Code (<https://www.aluracursos.com/blog/como-usar-el-terminal-integrado-de-visual-studio-code>)
- **Curso** Linux 1: conociendo y utilizando la terminal (<https://app.aluracursos.com/course/linux-1-conociendo-utilizando-terminal>)

Cloud - Fundamentos:

- La computación en nube, o cloud computing, es la distribución de servicios informáticos a través de Internet mediante un modelo de tarificación de pago por uso. Una nube se compone de varios recursos informatizados, desde los propios ordenadores (o instancias, en terminología de nube) hasta las redes, el almacenamiento, las bases de datos y todo lo que les rodea. En otras palabras, todo lo que normalmente se necesita para montar el equivalente a una sala de servidores, o incluso un centro de datos completo, estará listo para usar, configurar y ejecutar.
- Conocer la diferencia entre IaaS, PaaS y SaaS
- Conocer los mayores proveedores de nube
- Especializarse en un proveedor específico de su preferencia

Contenidos

- **Web** ¿Qué es la informática en la nube? | Microsoft Azure (<https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-cloud-computing/>)
- **Web** Amazon AWS: ¿Qué es la computación en nube? (<https://aws.amazon.com/en/what-is-cloud-computing/>)
- **Web** Tipos de computación en la nube (<https://aws.amazon.com/es/types-of-cloud-computing/>)
- **Web** ¿Cómo funciona Azure? (<https://learn.microsoft.com/es-es/azure/cloud-adoption-framework/get-started/what-is-azure>)
- **Web** ¿Qué es el almacenamiento en la nube? (<https://aws.amazon.com/es/what-is/cloud-storage/>)
- **Web** ¿Qué es la seguridad en la nube? (<https://cloud.google.com/learn/what-is-cloud-security?hl=es-419>)
- **Web** Arquitectura sin servidor (<https://learn.microsoft.com/es-es/dotnet/architecture/serverless/serverless-architecture>)
- **Web** ¿Qué es Docker? (<https://aws.amazon.com/es/docker/>)

- **Artículo** Guía para principiantes sobre los fundamentos de la computación en nube (<https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/>)
- **Artículo** Cloud Computing para principiantes (<https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/>)
- **Artículo** ¿Qué es Google Cloud y para qué sirve? (<https://www.incentro.com/es-ES/blog/que-es-google-cloud-platform>)
- **YouTube** ¿Qué es computación en la nube? | ¿Qué es cloud computing? | Explicado en 4 minutos (<https://youtu.be/MCKdahh2ISo>)
- **YouTube** 📌 CLOUD COMPUTING ¿Qué es IaaS, PaaS y SaaS? | Modelos de Servicio Cloud (<https://youtu.be/VR8aXePkQ5M>)
- **YouTube** ¿Qué es AWS? (<https://www.youtube.com/watch?v=x2vrg7HuM6g>)
- **YouTube** ¿Cómo empiezo con Google Cloud? (Hablemos en Cloud) (<https://www.youtube.com/watch?v=OiDWqu0oQfo>)
- **YouTube** Introducción a la infraestructura de Google Cloud (<https://www.youtube.com/watch?v=209DGQCism4>)
- **YouTube** ¿Qué es la Computación en la Nube? | AWS desde cero - Parte 1: Introducción (<https://www.youtube.com/watch?v=IciVhWQ8npw>)

Contenidos Alura:

- **Artículo** ¿Qué es Cloud y sus principales servicios? (<https://www.aluracursos.com/blog/que-es-cloud-y-sus-principales-servicios>)
- **Artículo** Conociendo Terraform (<https://www.aluracursos.com/blog/conociendo-terraform>)
- **Artículo** Empezando con Docker (<https://www.aluracursos.com/blog/empezando-con-docker>)
- **Artículo** Heroku, Vercel y otras opciones de cloud como plataforma (<https://www.aluracursos.com/blog/heroku-vercel-y-otras-opciones-de-cloud-como-plataforma>)
- **YouTube** Fundamentos del OCI | Contenidos ONE (<https://youtu.be/rEgSc0UqX-g>)
- **Curso** Curso Oracle Cloud Infrastructure: implementación de una aplicación en la nube (<https://app.aluracursos.com/course/oracle-cloud-infrastructure-implementacion-aplicacion-nube>)
- **Curso** Curso Oracle Cloud Infrastructure: base de datos e infraestructura como código (<https://app.aluracursos.com/course/oracle-cloud-infrastructure-base-datos-infraestructura-codigo>)

- **Curso** Curso Deploy en Amazon EC2: Alta disponibilidad y escalabilidad de una aplicación (<https://app.aluracursos.com/course/deploy-amazon-ec2-alta-disponibilidad-escalabilidad>)
- **Curso** Curso Amazon Lightsail: Simplificando la nube (<https://app.aluracursos.com/course/amazon-lightsail-simplificando-nube>)

SQL - Fundamentos:

- Conocer los comandos más comunes de SQL
- Usar SELECT para consultar una tabla
- Usar INSERT para insertar datos en una tabla
- Usar UPDATE para actualizar una tabla
- Usar DELETE para eliminar datos de una tabla
- Usar JOIN para conectar los datos de múltiples tablas
- Conocer las cláusulas (FROM, ORDER BY, etc.)

Contenidos

- **Artículo** Amazon: ¿Qué es SQL? (https://aws.amazon.com/es/what-is/sql/?nc1=h_ls)

Contenidos Alura:

- **Artículo** MySQL: desde la descarga e instalación hasta su primera tabla (<https://www.aluracursos.com/blog/mysql-desde-la-descarga-e-instalacion-hasta-su-primera-tabla>)
- **Artículo** Bases de datos relacionales (<https://www.aluracursos.com/blog/base-de-datos-relacional>)
- **Artículo** ¿Qué es SQL? (<https://www.aluracursos.com/blog/que-es-sql>)
- **Artículo** Normalización en base de datos - Estructura (<https://www.aluracursos.com/blog/normalizacion-en-base-de-datos>)
- **Artículo** En SQL, null es null, vacío está vacío (<https://www.aluracursos.com/blog/en-sql-null-es-null-vacio-es-vacio>)
- **Artículo** SELECT, INSERT, UPDATE y DELETE en SQL: aprende a utilizar cada uno (<https://www.aluracursos.com/blog/select-insert-update-delete-sql>)
- **Artículo** Funciones de agregación con GROUP BY en SQL, ¿cómo utilizarlas? (<https://www.aluracursos.com/blog/funciones-de-agregacion-con-group-by-en-sql-como-utilizarlas>)
- **Artículo** SQL JOIN: Aprenda INNER, LEFT, RIGHT, FULL e CROSS (<https://www.aluracursos.com/blog/sql-join-aprenda-inner-left-right-full-e-cross>)

- **Artículo** select count(*), count(1) y count(nombre): batalla de los counts de SQL (<https://www.aluracursos.com/blog/select-count-count1-e-countnome-la-batalla-de-los-counts-de-sql>)
- **YouTube** Descomplicando Base de Datos | #Aluramás (https://www.youtube.com/watch?v=G1cDRqKuxpg&t=6s&ab_channel=AluraLatam)
- **YouTube** ¿Qué es SQL y NoSQL? (<https://www.youtube.com/watch?v=cLLKVd5CNLc>)
- **YouTube** Banco de Datos MySQL (<https://www.youtube.com/watch?v=8J0AoPZMVxA>)
- **Curso** SQL con MySQL (<https://app.aluracursos.com/formacion-sql-con-mysql>)
- **Curso** Curso Introducción a SQL con MySQL: Manipule y consulte datos (<https://app.aluracursos.com/course/introduccion-sql-mysql-manipule-consulte-datos>)
- **Curso** SQL Server: consultas avanzadas con Microsoft SQL Server 2019 (<https://app.aluracursos.com/course/sql-server-consultas-microsoft-sql-server-2019>)
- **Curso** Formación Modelado de datos (<https://www.aluracursos.com/formacion-modelado-de-datos>)
- **Curso** Formación SQL con Microsoft SQL Server (<https://www.aluracursos.com/formacion-SQL-con-Microsoft-SQL-Server-2019>)

Habilidad Auxiliar: Buenas prácticas

SOLID:

- Solid tiene cinco principios considerados como buenas prácticas en el desarrollo de software que ayudan a los programadores a escribir los códigos más limpios, dividiendo las responsabilidades, disminuyendo los acoplamientos, facilitando la refactorización y estimulando el reaprovechamiento del código. Propuesto por Robert C. Martin, SOLID propicia el desarrollo de un código limpio, legible y comprobable.

Contenidos

- **Web** Los principios SOLID de programación orientada a objetos (<https://www.freecodecamp.org/espanol/news/los-principios-solid-explicados-en-espanol/>)
- **Web** Principios SOLID: Qué son, cuáles, y qué beneficios aporta usarlos (<https://devexperto.com/principios-solid/>)
- **Podcast** SOLID - los androides (<https://open.spotify.com/episode/47eElzhTPr1RluVcxeVal4?si=4df70e899c034283>)
- **YouTube** Los principios SOLID, ¡explicados! (<https://www.youtube.com/watch?v=2X50sKeBAcQ>)

Clean Architecture:

- Clean Architecture (Arquitectura Limpia) es una forma de desarrollar software, de tal forma que solo mirando el código fuente de un programa, debes ser capaz de decir lo que el programa hace.

Contenidos

- **Web** ¿Qué es Clean Architecture? (<https://clean-architecture-python.readthedocs.io/en/latest/introduccion/index.html>)
- **Artículo** ¿Por qué utilizo Clean Architecture? (<https://xurxodev.com/por-que-utilizo-clean-architecture-en-mis-proyectos/>)
- **YouTube** Revisando Clean code, vale la pena leerlo? | review clean code (<https://www.youtube.com/watch?v=uQfm6YaJTJl>)
- **YouTube** Hexagonal architecture, qué es y qué diferencias tiene contra Clean Architecture? - PT 1 (<https://www.youtube.com/watch?v=NOWU4K6piwo>)
- **YouTube** Desarrollo ágil con Arquitectura limpia Hexa3 (<https://medium.com/@dariopalminio/desarrollo-%C3%A1gil-de-ecosistemas-de-aplicaciones-hexagonales-3-capas-hexa3l-d6370bf11db0>)
- **Podcast** DevTalles - 125: Arquitectura Limpia (<https://open.spotify.com/episode/3ftlJfucj7Jx4ZO2cDSghx?si=e6b2c5607ead43b5>)

Contenidos Alura:

- **Artículo** Design Patterns: Breve introducción a los patrones de diseño (<https://www.aluracursos.com/blog/design-patterns-introduccion-a-los-patrones-de-diseno>)

Design Patterns:

- En ingeniería de software, un "patrón de diseño" (Design Pattern en inglés) es una solución general y reutilizable para un problema que ocurre normalmente dentro de un determinado contexto de diseño de software.
- Conocer y aplicar los principales patrones de diseño.

Contenidos

- **Web** Patrones de Diseño (<https://refactoring.guru/es/design-patterns>)
- **Web** Designer Patterns (<https://github.com/FernandoCalmet/design-patterns>)
- **Web** ¿Qué son los patrones de diseño de software? (<https://profile.es/blog/patrones-de-diseno-de-software/>)

- **Web** Design Patterns: conoce los diferentes tipos que existen y sus beneficios (<https://www.hostgator.mx/blog/design-patterns-que-debes-saber/>)
- **Podcast** Patrones Diseño, ventajas y desventajas (<https://open.spotify.com/episode/3VjQHnPVusU6zz5PyIMVFu?si=a613cb4c157e4055>)
- **Podcast** Patrones Diseño (<https://open.spotify.com/episode/6QO1HYdAgzrMGLVpz2kn0C?si=c95296f387c4490a>)
- **YouTube** ♦ Patrones de diseño software: Repaso completo en 10 minutos (<https://www.youtube.com/watch?v=6BHOeDL8vIs>)

Contenidos Alura:

- **Artículo** Design Patterns: Breve introducción a los patrones de diseño (<https://www.aluracursos.com/blog/design-patterns-introduccion-a-los-patrones-de-diseno>)

Clean Code:

- Aplicar técnicas sencillas para facilitar la escritura y la lectura de un código.
- Refactorizar el código para que quede más limpio.

Contenidos

- **Web** Clean Code: código limpio, ¿qué es? (<https://www.hostgator.mx/blog/clean-code-codigo-limpio/>)
- **Web** Clean Code: 35 claves para dominarlo (<https://devexperto.com/clean-code/>)
- **Podcast** ¿Que es Clean Code? - Programando Podcast (<https://open.spotify.com/episode/0xF40axFOYG1pdD2oaEMo0?si=b48fbfed5ec949c4>)
- **Artículo** Cómo escribir código limpio. Nombrando elementos (<https://medium.com/@latteandcode/como-escribir-codigo-limpio-nombrando-elementos-8caa6639602c>)
- **YouTube** Aprende a programar más limpio | The Clean Coder | Reseña del libro de Robert C. Martin (<https://www.youtube.com/watch?v=bZgWdELGBoc>)

Contenidos Alura:

- **Artículo** Clean Code: qué es, casos de uso, ejemplo de código limpio (<https://www.aluracursos.com/blog/clean-code-que-es-ejemplo>)

Diseño Orientado a Dominio - DDD:

- El Diseño Orientado a Dominio (DDD) es un enfoque de diseño y desarrollo de software que se informa principalmente por los requisitos de negocio. Los componentes del

programa (objetos, clases, matrices, etc.) indican la industria, sector o dominio empresarial en que opera el negocio.

- Modelar dominios de manera efectiva.
- Basar proyectos complejos en modelos de dominio.
- Conocer los bloques de construcción de DDD.

Contenidos

- **Web** Domain Driven Design: principios, beneficios y elementos — Primera Parte (<https://medium.com/@jonathanloscalzo/domain-driven-design-principios-beneficios-y-elementos-primera-parte-aad90f30aa35>)
- **Web** Domain Driven Design: principios, beneficios y elementos — Segunda Parte (<https://medium.com/@jonathanloscalzo/domain-driven-design-principios-beneficios-y-elementos-segunda-parte-337d77dc8566>)
- **Web** Introducción a Domain Drive Design - DDD (<https://refactorizando.com/introduccion-domain-drive-design/>)
- **Podcast** TDD y DDD, fortalezas y cuando usarlo (<https://open.spotify.com/episode/7iioBOv7ySQZttPyw2GshP?si=a114f642ecc94682>)
- **YouTube** Aprendiendo el Dominio (DDD en Español) (<https://www.youtube.com/watch?v=BilfTrk954A>)
- **YouTube** Domain Driven Design - Píldoras de conocimiento - Autentia (<https://www.youtube.com/watch?v=vccTIFGSiHA>)