

Master Java Online

Introducción

Actualmente, el lenguaje Java y las tecnologías que lo rodean constituyen un estándar de facto para el desarrollo de aplicaciones. Ya sean aplicaciones de escritorio, aplicaciones para entorno Web, incluso programas para dispositivos móviles, todos ellos pueden ser desarrollados utilizando el lenguaje Java y alguna de las librerías/frameworks que forman parte del ecosistema Java

Es especialmente en el campo de las aplicaciones Web donde Java, junto con la tecnología Java EE, son ampliamente utilizados tanto por compañías de desarrollo software como por programadores independientes.

Es por ello que el conocimiento de estas tecnologías constituye una garantía de éxito para todos aquellos que quieran desarrollar su carrera profesional en el mundo de la programación.

En este Master te proporcionamos la formación necesaria para que puedas trabajar como programador Java, utilizando las tecnologías y frameworks más demandados por el mercado. Comenzaremos con el estudio del lenguaje Java, sus características sintácticas, incluida la orientación a objetos, y todas las librerías de uso general. Abordaremos la creación de aplicaciones de escritorio y el acceso a bases de datos, como paso previo a adentrarnos en el mundo del desarrollo web.

Además de las tecnologías que forman Java EE para la construcción de aplicaciones Web, estudiaremos algunos de los frameworks más utilizados por la comunidad de desarrolladores, como JSF y Spring, así como la construcción de una capa de persistencia con JPA e hibernate. Para finalizar, abordaremos la arquitectura orientada a servicios, incluyendo la creación, despliegue y utilización de microservicios con las últimas tecnologías demandadas en el mercado.

Se trata de un master que cumple con todos los requisitos de lo que debe ser una formación de alto nivel para desarrollo de aplicaciones y cuyo único requisito académico para poder realizarlo es el conocimiento de algún lenguaje de programación.

Metodología Master Java on-line

Autoestudio

La modalidad económica de autoestudio, no incluye soporte de tutor, dado que dispone de numerosos recursos de aprendizaje con documentación, elementos multimedia, encuestas y evaluación final.

Planificación

Se define una planificación de 20 semanas para la realización del curso, en la que se indicará la temporización de cada módulo.

Tareas del alumno

Para todos los módulos, el alumno deberá seguir las siguientes pautas:

- Estudio de los contenidos en el orden en que aparecen en la plataforma
- Para cada ejercicio resuelto, el alumno lo probará en su entorno de desarrollo
- Cada ejercicio propuesto deberá ser realizado por el alumno en su entorno de desarrollo. Solo deberá consultar la solución para contrastar con lo realizado por él o después de haber intentado resolverlo por si mismo
- Ante cualquier duda, enviará una consulta al tutor a través de la plataforma.
- El tiempo de dedicación de estudio del curso será de 15 horas semanales.
- Al final de cada módulo realizará el test de evaluación que le permitirá obtener un feedback de su progreso
- Proyecto final (opcional) que puede ser un proyecto personal, la intención es que puedan llevarse algo tangible al acabar el curso.

Tareas del tutor

Durante el periodo de tutorización, el tutor debería realizar las siguientes tareas:

- Envío de email de bienvenida a cada alumno y puesta a disposición para la resolución de dudas
- Revisión diaria de consultas de alumnos, que se deberán resolver en un periodo máximo de 24 horas.
- Seguimiento de progreso de alumno y animación a los rezagados.
- *Java Café*. clases semanales de un par de horas en los que el tutor ayuda a los alumnos en sus proyectos personales (vía streaming con Tutor).

Documentación

Se generarán los siguientes documentos relativos al master:

- Descripción de contenidos de cada uno de los módulos
- Temporización de dedicación por parte del alumno a cada módulo del curso durante las 20 semanas
- Descripción de objetivos de cada módulo

Temario

Contenidos

Módulo 1. Programación en Java

- Introducción a la tecnología Java
 - Evolución y Características del lenguaje.
 - La JVM
 - Ediciones Java
- Software para trabajar en Java.
 - Descarga e instalación JDK
 - Descarga e instalación de Eclipse
 - video 1. Descargar e instalar eclipse
 - Creación y ejecución de un primer programa Java
- Sintaxis del lenguaje Java
 - Estructura de un programa Java
 - Tipos de datos, variables y constantes
 - Operadores
 - Ejercicio 1. División entre dos números
 - Instrucciones de control de flujo (if, switch,for, while)
 - Ejercicio 2. Mostrar mayor de dos números
 - Ejercicio 3. Resultado notas
 - Ejercicio 4. Suma pares
 - Ejercicio 5. Suma para mil
 - Arrays
 - Ejercicio 6. Almacenamiento 10 números y muestra datos
 - video 2. Utilización clase Scanner
- Clases de uso general
 - Concepto de clase, objeto y métodos
 - Ejercicio 7. Versión ejercicio 6 estructurada en métodos
 - Clase Math
 - Ejercicio 8
 - Gestión de cadenas con String y StringBuilder
 - Ejercicio 9
 - Manejo de clases de colección (listas, conjuntos y mapas)
 - Ejercicio 10
 - Clases de tipo JavaBean
 - Ejercicio 11
- Excepciones.
 - Concepto de excepción
 - Tipos
 - Captura de excepciones
 - Ejercicio 12
 - Try con recursos
- Testing de aplicaciones con JUnit

- Ejercicio 13

Módulo 2. Programación orientada a objetos

- Conceptos básicos
 - Encapsulación
 - Métodos de instancia y métodos estáticos
 - Ejercicio 14
 - Modificadores de acceso
 - Sobrecarga de métodos
- Constructores
 - Sintaxis
 - Ejercicio 15
 - Ejecución de un constructor
 - Sobrecarga de constructores
 - Constructores por defecto
 - Video 3. Presentación de ejemplo de varios constructores
- Herencia
 - Concepto y reglas de aplicación
 - Nomenclatura
 - Ejecución de constructores en la herencia
 - Ejercicio 16
 - Herencia en Java SE
 - Clases Finales
- Sobrescritura de métodos
 - Reglas para la sobrescritura de métodos
 - Ejercicio 17
 - Diferencias entre sobrescritura y sobrecarga
 - Métodos finales
- Clases abstractas
 - Métodos abstractos
 - Objetivo de una clase abstracta
 - Herencia de clases abstractas
 - Ejercicio 18
- Polimorfismo
 - Concepto y puesta en práctica
 - Ventajas del polimorfismo
 - Ejercicio 19
 - El polimorfismo en el API Java SE
- Interfaces
 - Concepto y definición
 - Polimorfismo en interfaces
 - Ejercicio 20
 - Interfaces en Java 8 y 9
 - Ejercicio 21
 - Interfaces funcionales

- Expresiones lambda y su aplicación
 - Concepto
 - Reglas de construcción
 - Video 4. Sobre expresiones lambda
 - Implementación de interfaces funcionales con lambdas
 - Ejercicio 22
 - Nuevas interfaces funcionales Java 8
 - Nuevos métodos en colecciones de Java 8
 - Ejercicio 23
- Streams
 - Características de los Streams
 - La interfaz Stream
 - Operaciones intermedias y finales
 - Ejercicio 24
 - Aplicación de métodos más destacados de Streams
 - Ejercicio 25

Módulo 3. Creación de aplicaciones para Java estándar

- Entrada y salida en Java
 - El paquete java.io
 - Escritura con PrintStream
 - Escritura de datos con formato
 - Lectura de datos por teclado con Scanner
 - Lectura de datos por teclado con BufferedReader
- Acceso a ficheros
 - Escritura de datos en un fichero en modo sobrescritura
 - Escritura de datos en un fichero en modo adición
 - Lectura secuencia de un fichero con BufferedReader
 - Introducción al paquete java.nio.files
 - Lectura de ficheros con Files
- Multitarea
 - Concepto de multitarea
 - Aplicaciones multitarea con Thread
 - Aplicaciones multitarea con Runnable
 - Sincronización en el acceso a recursos
 - Creación de aplicaciones multitarea con java.util.concurrent
- Comunicaciones en Java
 - Fundamentos de comunicaciones
 - Acceso a recursos Web con HttpURLConnection
 - La clase InetAddress
 - Conexión mediante sockets
 - Creación de sockets de servidor con ServerSocket
- Aplicaciones basadas en entorno gráfico
 - Introducción al paquete javax.swing
 - Creación y configuración de una ventana JFrame
 - Creación e inclusión de componentes gráficos en una ventana
 - Gestión de eventos en swing
 - Estudio de controles básicos
 - Listas y tablas
 - Cuadros de diálogo
- Acceso a base de datos en Java
 - Fundamentos de bases de datos y SQL
 - El API JDBC
 - Utilización de drivers
 - Conexión a una base de datos
 - Ejecución de consultas de acción
 - Manipulación de resultados con ResultSet
 - Consultas preparadas
 - Ejecución de procedimientos almacenados
 - Transacciones

Módulo 4. Desarrollo de aplicaciones Web con Java

- Arquitectura de aplicaciones Web
 - La arquitectura de tres capas
 - Tecnologías implicadas en cada capa
 - Servidores de aplicaciones y servidores Web
- Introducción a Java EE
 - Componentes y servicios Java EE
 - Estructura de una aplicación Web
 - Formatos de despliegue de aplicaciones
 - Contenedores y servidores de aplicaciones Java EE
- Programación con servlets
 - Creación de un servlet y ciclo de vida
 - Generación de respuestas.
 - Recogida de parámetros de usuario
 - Transferencia y redireccionamiento de peticiones
 - Atributos de petición, sesión, aplicación y cookies
 - Eventos y escuchadores
 - Filtros
- Creación de páginas Java Server Pages (JSP)
 - Creación de una JSP
 - Relación entre JSP y Servlets
 - Objetos implícitos
 - Directivas JSP
 - Acciones JSP
 - El lenguaje EL y la librería JSTL core
- El patrón Modelo Vista Controlador
 - Esquema del patrón de diseño MVC
 - Descripción de cada bloque
 - Acceso a datos en aplicaciones Web
 - Análisis del controlador
 - Desarrollo de una aplicación completa siguiendo este patrón
 - Utilización de DataSources en acceso a datos
- Desarrollo front end
 - Inserción de scripts en una página cliente
 - El lenguaje JavaScript
 - Principales objetos JavaScript
 - Técnica de programación AJAX
 - Utilización de la librería Jquery
 - Manipulación de datos JSON

Módulo 5. Componentes EJB y persistencia con JPA

- Introducción a la arquitectura Enterprise JavaBeans
 - Definición y características de un EJB
 - Contenedores EJB y servicios que ofrece
 - Tipos de EJB
- Implementación de un EJB stateless
 - EJB en aplicaciones Web
 - Creación de un EJB stateless
 - Interfaz de negocio y clase de implementación
 - Acceso a un EJB desde un cliente servlet
- Otros tipos de EJB
 - Creación de EJB statefull
 - EJB singleton
 - EJB de tipo MDB
 - Temporizadores
- Capa de persistencia en una aplicación
 - Introducción a la persistencia. Características
 - Java Persistence API (JPA)
 - Creación de una capa de persistencia
 - Primera aplicación JPA
- Operaciones básicas con EntityManager
 - Obtención de EntityManager
 - Realización de operaciones CRUD
 - Transaccionalidad
 - Integración de JPA en EJB
- Consultas JPA
 - El lenguaje JPQL
 - Creación de una consulta
 - Consultas parametrizadas
 - Consultas nominadas
 - Consultas de acción
- Relaciones entre entidades
 - Tipos de relaciones
 - Definir entidades relacionadas
 - Ajuste de relaciones
 - Relaciones entre entidades con claves compuestas
- Integración con Hibernate
 - Características de hibernate
 - Utilización de motor de hibernate con API JPA
 - Configuración de dependencias con Maven

Módulo 6. El framework MVC JSF

- Primeros pasos con JSF
 - Introducción a JSF
 - Componentes de controlador y vista en JSF
 - Dependencias JSF con Maven
 - Configuración del entorno de desarrollo
 - Primera aplicación JSF
- Funcionamiento de JSF
 - La vinculación de componentes gráficos a managed beans
 - Componentes gráficos básicos
 - Ciclo de vida de una petición JSF
 - Inyección de beans
 - El API JSF
- Componentes gráficos de tipo lista y tabla
 - Tipos de listas
 - Generación estática y dinámica de una lista
 - Tablas
- Integración de JSF con otras tecnologías
 - Integración con EJBs
 - Integración con Spring
 - AJAX y JSF
 - Plantillas JSF
- Validación de datos en JSF
 - Estudio de los principales validadores
 - Creación de validadores personalizados
 - Internacionalización de aplicaciones con JSF
- Componentes avanzados
 - Librerías de componentes externos
 - La librería Primefaces
 - Análisis de componentes Primefaces

Módulo 7. Spring

- Fundamentos de Spring
 - Introducción y ámbito de utilización de Spring
 - Módulos que forman el framework
 - Inclusión de básicas en un proyecto
- Primeros pasos con Spring
 - Características del módulo Core
 - Inyección de dependencias
 - Configuración e instanciación de beans
 - Configuración por XML y mediante clases
 - Acceso a beans des Spring desde una aplicación
- Inyección de beans
 - Opciones de configuración de beans
 - La anotación @Autowired
 - Autoconfiguración de beans con @Component

- Utilización de DataSources
- Acceso a datos con Spring Dao
 - Dependencias de Spring Dao
 - Utilización de JdbcTemplate
 - Principales métodos para acceder a datos
 - Integración de Spring con JPA
 - Transaccionalidad
- Orientación a Aspectos con Spring
 - Características de la programación orientada a aspectos
 - Nomenclatura
 - Utilización de AspectJ con Spring. Dependencias del proyecto
 - Creación y configuración de aspectos
- Spring MVC
 - Integración de Spring con otros frameworks MVC
 - Características del módulo Spring MVC
 - Dependencias en el proyecto y configuración
 - Implementación de controladores
 - Resolución de vistas
 - Acceso a datos de un formulario
 - Librerías para JSP
- Spring security
 - La seguridad Java EE
 - Dependencias para Spring Security
 - Anotaciones para la capa del modelo
 - Seguridad en la capa Web
 - Repositorios de usuarios
- Spring boot
 - Características de Spring boot
 - Dependencias del proyecto
 - Creación de un proyecto autoconfigurable

Módulo 8. Servicios Web

- Arquitectura orientada a servicios
 - Principios de SOA
 - Concepto de Servicio Web
 - Roles en la arquitectura de servicios Web
 - Tipos de servicios
- Servicios Web XML
 - Funcionamiento y características
 - Tecnologías en la arquitectura de servicios Web XML
 - Implementación de servicios Web XML con Java
 - El API JAX-WS
 - Aplicaciones clientes de servicios Web
 - Acceso mediante proxy y acceso directo
 - Acceso asíncrono a servicios Web

- Acceso a servicios Web reales
- Servicios Web de tipo REST
 - Características de los servicios REST
 - Implementación de servicios REST
 - El API JAX-RS
 - Utilización de Jersey en servidor
 - Creación de aplicaciones clientes REST
 - Utilización de Jersey en cliente
 - Acceso a servicios REST reales
- Creación de microservicios con Spring
 - Concepto de microservicio
 - Creación de microservicios con Spring Rest
 - Configuración y despliegue de microservicios con Spring boot
 - Aplicaciones clientes de microservicios con Spring
 - Configuración distribuida en Spring Cloud
- Integración de servicios en aplicaciones Web
 - Definición de los servicios
 - Interconexión entre servicios
 - Despliegue de servicios
 - Acceso desde aplicaciones Web
 - Acceso a servicios desde front-end

Objetivos globales del master

- Conocer las diferentes estructuras sintácticas del lenguaje Java y su correcta aplicación en la construcción de aplicaciones.
- Comprender los diferentes conceptos sobre orientación a objetos y aplicarlos adecuadamente en los programas.
- Identificar las principales clases de Java estándar y los métodos más importantes de las mismas.
- Aplicar las diferentes clases de Java estándar para construir aplicaciones de escritorio.
- Conocer y aplicar las tecnologías y componentes de Java enterprise edition en la construcción de aplicaciones Web.
- Aplicar los patrones y buenas prácticas para la creación de aplicaciones Web eficientes y escalables.
- Utilizar las tecnologías Java basadas en persistencia para acceder a datos en aplicaciones, así como su integración con el resto de capas del programa.
- Conocer los principales frameworks utilizados por las empresas de desarrollo para la creación de aplicaciones de forma rápida, segura y eficiente.
- Identificar los principales elementos y tecnologías utilizadas en la arquitectura de servicios Web y su empleo en la construcción y despliegue de este tipo de componentes, así como de aplicaciones clientes de los mismos.

Objetivos de cada módulo

Módulo 1. Programación en Java

- Identificar las características y ventajas que ofrece la tecnología Java a la hora de crear aplicaciones.
- Manejar el entorno de desarrollo eclipse para la creación de aplicaciones Java estándar
- Conocer los tipos de datos primitivos y su manejo a través de variables
- Utilizar las estructuras de control de Java para definir el flujo de un programa
- Conocer y manejar arrays en un programa
- Conocer las clases básicas de uso general en un programa Java y sus principales métodos
- Aplicar el mecanismo de excepciones para el control de errores en un programa

Módulo 2. Programación orientada a objetos

- Identificar los componentes de una clase y el efecto de aplicar los diferentes modificadores de acceso.
- Aplicar correctamente la sobrecarga, tanto en métodos como constructores
- Identificar los beneficios del uso de la encapsulación
- Aplicar correctamente la herencia en la creación de clases y comprender los beneficios del uso de la misma
- Aplicar correctamente la reglas de la sobrescritura de métodos
- Conocer las clases abstractas, su uso y aplicación
- Identificar los beneficios del uso del polimorfismo en la creación de aplicaciones
- Conocer las interfaces, sus componentes y su aplicación, identificado las interfaces de tipo funcional
- Utilizar las expresiones lambda en la implementación de interfaces
- Conocer y aplicar los principales métodos de streams para la manipulación de grupos de datos

Módulo 3. Creación de aplicaciones Java Estándar

- Identificar las principales clases del paquete javax.swing
- Crear ventanas y componentes gráficos en una aplicación
- Gestionar eventos en una aplicación gráfica
- Conocer y manejar listas y tablas para la presentación y captura de datos en una aplicación
- Utilizar cuadros de diálogo para la interacción con el usuario dentro de una aplicación

Módulo 4. Desarrollo de aplicaciones Web con Java EE

- Conocer la arquitectura en tres capas
- Analizar las características y ventajas de las aplicaciones web
- Instalar y configurar un servidor de aplicaciones
- Aprender a programar con Servlets y JSP
- Utilizar JSTL para implementar la lógica de las páginas JSP

- Conocer y aplicar el patrón MVC en la construcción de aplicaciones
- Aprender a trabajar con Datasources

Módulo 5. Desarrollo de componentes EJB y persistencia con JPA

- Identificar las características de los EJBs y los principales servicios proporcionados por un contenedor
- Identificar los diferentes tipos de EJB y sus peculiaridades
- Utilizar adecuadamente los distintos tipos de EJBs para la implementación de la capa de negocio en una aplicación
- Conocer las diferentes formas de acceder a un EJB desde una aplicación cliente
- Comprender las características y ventajas de la utilización de una capa de persistencia
- Aplicar los diferentes métodos de EntityManager en la realización de operaciones básicas
- Conocer el lenguaje JPQL y su uso para creación de consultas JPA
- Aplicar los mecanismos para establecer relaciones entre entidades

Módulo 6. El framework MVC JSF

- Identificar los principales componentes del framework JSF
- Implementar adecuadamente Managed Beans en la creación del controlador
- Comprender el ciclo de vida de una petición JSF
- Utilizar las técnicas para inyección de objetos en una aplicación JSF, tanto objetos externos como otros managed beans
- Conocer los principales componentes gráficos JSF y su vinculación a Managed Beans
- Aplicar las técnicas para la validación de datos en una aplicación mediante JSF
- Utilizar librerías de componentes externos para la creación de interfaces gráficas avanzadas

Módulo 7. Spring

- Comprender el funcionamiento del core de Spring y la manera en la que crea y gestiona instancias de beans
- Aplicar la técnica de inyección de dependencia para la obtención de objetos Spring
- Identificar y aplicar adecuadamente las principales anotaciones de Spring Core
- Conocer los componentes de Spring DAO para acceder a datos en aplicaciones
- Aplicar las técnicas de orientación de aspectos mediante Spring y AspectJ para la creación de funcionalidades transversales
- Utilizar los componentes de Spring MVC en la construcción de aplicaciones Web
- Aplicar las técnicas y componentes para la integración de otros frameworks en Spring
- Conocer y aplicar los principales elementos de Spring security para la securización de aplicaciones, tanto en la capa Web como en la capa del modelo

Módulo 8. Servicios Web

- Comprender las características de la arquitectura orientada a servicios y las ventajas del uso de servicios Web
- Identificar las tecnologías de la arquitectura de servicios Web
- Conocer el API JAX-WS y emplearlo en la creación de servicios Web y de aplicaciones clientes de servicios

- Comprender el funcionamiento de los servicios REST
- Aplicar el API JAX-RS en la creación de servicios REST y de aplicaciones clientes de estos servicios
- Utilizar el API Spring REST para la creación de servicios y su despliegue con Spring Boot
- Comprender el concepto de microservicio los mecanismos para la implementación y despliegue de los mismos a través de Spring.