

## TEMARIO CURSO SPRING WEB SERVICES

- Instalación y configuración de Java JDK
- Instalación Eclipse
- Instalación y configuración de Maven
- Spring tools
  - Integrar JDK
  - Integrar Maven
- Crear una aplicación con Spring Boot
  - Explicación de ArtifactId
  - Explicación de GroupId
  - Librería Spring Boot DevTools
  - Librería Spring WEB
- Estructura de una aplicación Spring Boot
- Anotaciones Basicas
  - Anotación SpringBootApplication
  - Anotación RestController
  - Anotación RequestMapping
  - Anotación GetMapping
- Inyección de Dependencia
  - Anotación @Value
  - Anotación Component
  - Inyección por Atributo
  - Anotación Autowired
  - Objeto ConfigurableApplicationContext
  - Método getBean()
  - Inyección por Constructor
- Stereotypes
  - Anotación @Component
  - Anotación @Service
  - Anotación @Controller
  - Anotación @Repository
- Cualificadores
  - Anotación @Qualifier
  - Utilizando interfaz
  - Anotación @Primary
    - A nivel de Component
- Perfiles
  - Anotación @Profile
    - Tiempo de Desarrollo
    - Tiempo de Producción
    - Tiempo de QA
  - Activar el Perfiles

- Mediante Archiv application.properties
- Mediante Virtual Machine
- Mediante Anotación @ActiveProfile
- Ciclo de Vida de Spring
  - Scopes
  - Anotación @Scope
    - Singleton
    - Prototype
    - Request
    - Session
- Beans Explicitos
  - Anotación @Bean
- Inyección Múltiples Objetos
  - Utilizando Listas
- Reemplazar Anotación @SpringBootApplication
  - Anotación @Configuration
  - Anotación @ComponentScan
  - Anotación EnableAutoConfiguration
- Carga de Properties
  - Archivo .properties
  - Anotación @PropertySource
    - Parametro classpath
- SPEL en Spring
  - Uso de \${}
  - Objeto ExpressionParser
  - Operadores Matemáticos
    - +, -, \*, /
  - operadores relacionales
    - <, >, ==, !=, <=, >=, lt, gt, eq, ne, le, gt
  - Agregar operadores logicos
    - And, or, &&, ||
- Ciclo de Vida
  - Awares
    - BeanNameAware
    - BeanFactoryAware
  - Callbacks
    - Anotación @PostConstructor
    - Anotación @PreConstructor
  - Anotación @Bean
    - Propiedad initMethod
    - Propiedad destroyMethod
  - Interfaces
    - InitializingBean

- DisposableBean
- Inicialización
  - BeanEager
  - BeanLazy
- Interfaz BeanPostProcessor
  - Método postProcessBeforeInitialization()
  - Método postProcessAfterInitialization()
- Orden de Ejecución
- Programación Orientada a Aspectos AOP
  - Aspectos
  - JoinPoint
  - Advice
    - Before
    - After Returning
    - After Throwing
    - After Finally
    - Around
  - PointCut
  - TargetObject
  - AOP Proxy
  - Weaving
  - Dependencias para AOP
- Precedencia de Pointcuts
  - Anotación @Order
- Reutilizar Pointcuts
  - Anotación @Pointcut
- Type Signatures Patterns
  - Utilizando withing()
- Crear nuestras propias anotaciones
  - Anotación @annotation
  - Crear una interfaz
- HTTP Protocolo
- Metodos HTTP
  - Método get
  - Método Head
  - Metodo Post
  - Método Put
  - Método Delete
  - Método Trace
  - Método Connect
- Petición HTTP
  - URI
  - Version

- Headers
- Body
- Respuesta HTTP
  - Version
  - Status
  - Headers
  - Body
- Status HTTP
- Spring Rest
  - Configuración
  - Crear Proyecto WEB
  - Anotaciones
    - Anotación @Controller
    - Anotación @RestController
    - Anotación @RequestMapping
    - Anotación @GetMapping
    - Anotación @PostMapping
    - Anotación @PutMapping
    - Anotación @DeleteMapping
    - Anotación @PathVariable
    - Anotación @RequestParam
  - Configurar JavaFaker
  - Crear Registros con JavaFaker
  - Consultar Registros con JavaFaker
  - Consultar registros por Id
- Instalación de Postman
- Crear Usuarios
  - Objeto ResponseStatusExeption
  - Anotación @PostMapping
  - Anotación @RequestBody
- Actualizar Usuarios
  - Anotación @PutMapping
- Eliminar Usuarios
- Parametros en URL
  - Param Querys
- Java Persistence API
  - Utilizando H2
- Spring Data
  - ORM
    - Anotación @Entity
    - Anotación @Table
    - Anotación @Id
    - Anotación @GeneratedValue

- Anotación @Column
- Relaciones
  - Anotación @OneToOne
  - Anotación @ManyToOne
  - Anotación @ManyToMany
- Dependencias spring boot jpa
- Dependencia H2
- Configurar .properties con H2
- Tipos de Repositorios
  - CrudRepository
    - Método save()
    - Método findOne()
    - Método findAll()
    - Método count()
    - Método Remove()
    - Método exists()
  - PagingAndSortingRepository
  - JpaRepository
    - Método findAll()
    - Método flush()
    - Método saveAndFlush()
    - Método deleteInBatch()
- Query Methods
  - Agregar métodos a la interface repository
    - Método findByUsername()
    - Método findByUsernameAndPassword()
  - Ruta para obtener por Id
  - Ruta para obtener por nombre de usuario
  - Ruta para obtener usuario y pass
- Paginación
  - Objeto PageRequest
    - Propiedad page
    - Propiedad size
  - Método of()
- Querys personalizados
  - Anotación @Query
- Actuator
  - Configuración de Actuator
- Metricas
  - Crear Metricas
  - Prometheus
  - Anotación Timed
- Grafana - Visualizar Datos

- Instalación
- Integrar prometheus con Grafana
- Swagger para Documentar
  - Dependencias a utilizar
  - Anotación @ApiOperation
  - Anotación ApiResponse
- Spring Caching
  - Almacenar datos en cache
  - Anotación @Cacheable
  - Borrar la cache
  - Anotación @CacheEvict

