«Talento Tech»

Back-End

Java

Clase 10

Buenos Aires Ciudad

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Clase 10: Introducción a Spring Boot

Índice

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Objetivos de la Clase

- Comprender el lugar de Spring como framework central para aplicaciones Java, y la función de Spring Boot como facilitador.
- Conocer las ventajas de usar Spring Boot: autoconfiguración, servidor embebido, integración con el ecosistema Spring.
- Aprender a crear un proyecto Spring Boot desde cero utilizando herramientas como Spring Initializr.
- Familiarizarse con la estructura básica de un proyecto Spring Boot, las dependencias esenciales y los archivos de configuración.
- Dar los primeros pasos implementando un endpoint REST simple con Spring Boot.

Bienvenida a TechLab

Talento Lab

¡Les damos la bienvenida nuevamente a TechLab! Hasta ahora, hemos construido una sólida base con Java, POO, manejo de excepciones, organización del código, control de versiones con Git y un backend conceptual. Sin embargo, el proyecto de TechLab crece y exige un entorno que facilite la creación de APIs web, el acceso a bases de datos, la seguridad y la escalabilidad. Esto nos lleva a Spring Boot, un framework que simplifica el desarrollo de aplicaciones Java modernas, y al ecosistema Spring, un conjunto de proyectos orientados a resolver problemas comunes en el desarrollo empresarial.
Con Spring Boot, podrán implementar servicios web productivos con mínima configuración, integrando rápidamente componentes como controladores REST, seguridad, acceso a datos y otras funcionalidades avanzadas con apenas unas dependencias y anotaciones.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

¿Qué es Spring y su Ecosistema?

Spring es un ecosistema de proyectos orientados a simplificar el desarrollo de aplicaciones Java empresariales. Nació como un framework para manejo de dependencias e inversión de control, ofreciendo inyección de dependencias (IoC) y facilitando el desarrollo modular y desacoplado. Con el tiempo, se expandió para cubrir múltiples áreas:

Spring Framework

El Spring Framework es la base. Ofrece inyección de dependencias, programación orientada a aspectos (AOP), y una infraestructura para desarrollar aplicaciones Java robustas. Antes de Spring, gran parte de la configuración era manual, pero Spring introdujo la idea de IoC contenedor, anotaciones y configuración declarativa, lo que simplificó enormemente el desarrollo.

El Ecosistema Spring: Spring Data, Spring Security, Spring MVC, etc.

- Spring MVC: Facilitó la creación de aplicaciones web y controladores para endpoints HTTP.
- Spring Data: Simplifica el acceso a datos, integrando repositorios CRUD y abstrayendo el acceso a bases relacionales y no relacionales.
- Spring Security: Proporciona autenticación, autorización y otras características de seguridad de forma simple y extensible.
- Spring Integration, Spring Batch, Spring Cloud: Extienden el ecosistema a mensajería, procesamiento por lotes, integración con servicios en la nube y más.

El ecosistema Spring es amplio y modular. Podés elegir solo las piezas que necesitás.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

¿Qué es Spring Boot y por qué Utilizarlo?

Spring Boot es un proyecto que nace para simplificar radicalmente la creación de aplicaciones basadas en Spring. Antes de Spring Boot, configurar una aplicación Spring requería muchos archivos XML, configuraciones manuales y decisiones complejas. Spring Boot automatiza gran parte de esta configuración usando el principio de convención sobre configuración.

Convención sobre Configuración

En lugar de configurar todo manualmente, Spring Boot asume configuraciones razonables por defecto. Por ejemplo, si incluís la dependencia spring-boot-starter-web, Spring Boot asume que querés un servidor web embebido (Tomcat por defecto) y configura automáticamente Spring MVC.

Servidor Embebido y Startups Rápidos

Con Spring Boot, no necesitas desplegar tu aplicación en un servidor externo. El proyecto se ejecuta con java -jar, y el servidor (Tomcat, Jetty o Undertow) se inicia automáticamente. Esto agiliza el desarrollo, ya que podés probar tu API inmediatamente después de un commit.

Actuadores, Monitorización y Facilidades Operacionales

Spring Boot Actuator ofrece endpoints para monitorear la salud, las métricas y la información del sistema. Facilita la integración con herramientas de monitoreo y alertas. Además, su integración con Spring Cloud, Kubernetes u otras infraestructuras modernas es muy fluida.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Creación de un Proyecto Spring Boot desde Cero

La forma más sencilla es usar el Spring Initializr (start.spring.io), una herramienta web:

1. Ingresar a start.spring.io
2. Elegir versión de Spring Boot, nombre del grupo (com.techlab), artefacto (backend), el lenguaje (Java) y el Java version.
3. Seleccionar dependencias como Spring Web (para crear APIs REST), Spring Data JPA (para acceso a base de datos), etc.
4. Descargar el proyecto generado y abrirlo en tu IDE.

Estructura Básica del Proyecto

Un proyecto Spring Boot típico se compone de:

- src/main/java: Código fuente del backend.
- src/main/resources: Archivos de configuración (application.properties / application.yml), plantillas, estáticos.
- pom.xml (Maven) o build.gradle (Gradle): Archivo de dependencias y configuración de compilación.
- Clases anotadas con @SpringBootApplication inician la aplicación.

Dependencias Comunes

- spring-boot-starter-web: Para crear endpoints REST.
- spring-boot-starter-data-jpa: Para acceso a datos con JPA.
- spring-boot-starter-security: Para seguridad en el backend.
- spring-boot-starter-test: Para pruebas unitarias e integración.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Configuración Inicial y Dependencias Necesarias

La configuración se gestiona en application.properties o application.yml. Podés definir el puerto del servidor (server.port=8080), propiedades de la base de datos, entre otros. Para entornos diferentes, podés usar perfiles (spring.profiles.active=desarrollo).
La configuración de dependencias se realiza en pom.xml o build.gradle. Añadir una dependencia es tan simple como agregarla al archivo y luego mvn install o gradle build.

Primeros Pasos con Spring Boot

Un ejemplo básico:

```
@SpringBootApplication
public class TechlabApplication {
    public static void main(String[] args) {
        SpringApplication.run(TechlabApplication.class, args);
    }
}
Esta clase arranca la aplicación. Con spring-boot-starter-web, podés
crear un controlador:
java
Copy code
@RestController
@RequestMapping("/productos")
public class ProductoController {
    @GetMapping
    public List<Producto> listarProductos() {
        // Retorna lista de productos (ejemplo)
        return Arrays.asList(new Producto("Café", 250.0, 100));
    }
}
```

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Al ejecutar la aplicación (mvn spring-boot:run o ./gradlew bootRun), podrás acceder a http://localhost:8080/productos y ver la lista de productos.

Inyección de Dependencias y Uso de @Autowired

Con Spring Boot y Spring, podés inyectar dependencias mediante anotaciones como @Autowired en constructores o atributos. Por ejemplo:

```
@Service
public class ProductoService {
    // lógica para manejar productos
}

@RestController
public class ProductoController {
    private final ProductoService productoService;

    @Autowired
    public ProductoController(ProductoService productoService) {
        this.productoService = productoService;
    }
}
```

Spring Boot detecta @Service, @Controller, @Repository y otras anotaciones, registrando beans en el contenedor IoC y resolviendo dependencias automáticamente.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Situación Inicial en TechLab

Silvia, la Product Owner, quiere exponer la lógica del backend a un frontend o a clientes externos. Matías y Sabrina han oído hablar de Spring Boot y cómo acelera la creación de APIs REST. Buscan un framework maduro, con buena documentación, amplio soporte en la industria y extensible para necesidades futuras.

El equipo nota que con Spring Boot pueden arrancar un servidor en minutos, exponer endpoints, conectarse a una base de datos y añadir seguridad progresivamente. Esto reducirá el tiempo de configuración manual y les permitirá centrarse en la lógica del negocio, justo lo que TechLab necesita para escalar y ofrecer servicios estables.

Ejercicios Prácticos

1. Crear un Proyecto Spring Boot con Spring Initializr:
    - Selecciona Spring Web y Spring Data JPA.
    - Descarga el proyecto, importalo en tu IDE.
    - Agregá un endpoint GET /hello que devuelva un mensaje simple.
2. Configurar un Puerto Diferente:
    - En application.properties, pon server.port=9090.
    - Verificá que la aplicación corra en http://localhost:9090.
3. Agregar un Servicio e Inyectarlo:
    - Creá un ProductoService que devuelva una lista de productos.
    - Inyectalo en ProductoController y usá productoService para servir datos al endpoint.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Materiales y Recursos Adicionales

- Documentación Oficial de Spring:
- Guía de Spring Boot:
- Videos recomendados en YouTube sobre Spring Boot, configuración inicial y creación de endpoints REST.

Preguntas para Reflexionar

- ¿Cómo se integra Spring Boot con el ecosistema Spring para agilizar el desarrollo?
- ¿Qué ventajas ofrece el arranque rápido y el servidor embebido frente a los métodos tradicionales de despliegue de aplicaciones Java?
- ¿De qué manera la autoconfiguración y las convenciones de Spring Boot reducen la carga cognitiva del equipo en TechLab?

Próximos Pasos

En la próxima clase, profundizaremos en la comunicación con bases de datos usando Spring Data, o avanzaremos en el desarrollo de APIs más complejas con seguridad y persistencia. Mientras tanto, practicá crear proyectos, añadir endpoints y explorar las posibilidades de configuración y dependencias con Spring Boot. ¡Felicitaciones por llegar hasta aquí! Con Spring Boot, TechLab cuenta con una herramienta poderosa para crear servicios escalables, robustos y listos para producción, integrados con el vasto ecosistema Spring.

Buenos Aires Ciudad