Roadmap para el desarrollo con Spring Boot

Este roadmap cubre las tecnologías esenciales y herramientas para el desarrollo, despliegue y monitoreo de microservicios.

Fase 1: Fundamentos de Java y Programación Orientada a Objetos (POO)

1. **Java Básico:**
   - Sintaxis básica, estructuras de control, colecciones, manejo de excepciones.
   - Programación Orientada a Objetos (POO): Clases, objetos, herencia, polimorfismo, interfaces, encapsulación.
   - Java I/O: Lectura y escritura de archivos, serialización.
   - Conceptos avanzados: Streams, expresiones lambda, manejo de concurrencia, multithreading.
2. **Java Avanzado:**
   - Generics, colecciones avanzadas (HashMap, LinkedHashMap, TreeMap, etc.).
   - Anotaciones, Reflection.
   - Programación funcional con Java.
   - Manejo de concurrencia con `java.util.concurrent`.

Fase 2: Fundamentos de Spring Framework y Spring Boot

1. **Spring Core:**
   - Fundamentos del framework Spring: Inversión de Control (IoC) y la inyección de dependencias (DI).
   - Configuración de Spring (XML vs Java Config).
   - Ciclo de vida de los beans y ámbito de los beans (`@Component`, `@Service`, `@Repository`, `@Controller`).
2. **Spring Boot:**
   - Configuración de un proyecto Spring Boot.
   - Uso de `Spring Initializr`.
   - Auto-configuración y conveniencia de Spring Boot.

- Archivos de configuración (`application.properties`, `application.yml`).
- Spring Boot Starters.

3. **Spring Data JPA:**
   - Integración con bases de datos relacionales usando Spring Data JPA.
   - Repositorios (CrudRepository, JpaRepository).
   - Definición de entidades, mapeo de relaciones (OneToMany, ManyToMany).
4. **Spring MVC:**
   - Controladores REST (`@RestController`, `@GetMapping`, `@PostMapping`, etc.).
   - Manejo de excepciones (`@ControllerAdvice`).
   - Validación de datos (`@Valid`, `@NotNull`, etc.).
5. **Spring Security:**
   - Autenticación y autorización con Spring Security.
   - Configuración de seguridad básica.
   - JWT (JSON Web Tokens) y OAuth2.

Fase 3: Arquitectura de Microservicios

1. **Principios de Microservicios:**
   - Comprender los principios de microservicios: Desacoplamiento, modularidad, escalabilidad, autonomía de servicios.
   - Diseño orientado a dominios (DDD) y patrones como CQRS (Command Query Responsibility Segregation).
   - Patrones de arquitectura de microservicios: API Gateway, Service Registry, Circuit Breaker, etc.
2. **Desarrollo de Microservicios con Spring Boot:**
   - Creación de microservicios independientes con Spring Boot.
   - Uso de Spring Cloud para la integración de microservicios.
   - Comunicación entre microservicios (REST, gRPC, Kafka, etc.).
   - Manejo de transacciones distribuidas con patrones como Saga.
3. **API Gateway y Service Registry:**
   - Configuración de un API Gateway (Spring Cloud Gateway o Zuul).
   - Implementación de un Service Registry (Eureka o Consul).
4. **Patrones de Resiliencia:**

- Implementación de patrones de resiliencia: Circuit Breaker (Hystrix, Resilience4j), Bulkhead, Retry.
- Uso de Spring Cloud Circuit Breaker para mejorar la resiliencia.

Fase 4: Bases de Datos y Persistencia

1. **Bases de Datos Relacionales y NoSQL:**
   - Fundamentos de bases de datos relacionales (PostgreSQL, MySQL).
   - Fundamentos de bases de datos NoSQL (MongoDB, Cassandra).
2. **ORM y JPA:**
   - Uso avanzado de JPA y Hibernate.
   - Optimización de consultas, caching de entidades, gestión de transacciones.
3. **Bases de Datos en Microservicios:**
   - Estrategias de partición de bases de datos y sincronización.
   - Event Sourcing y CQRS.

Fase 5: Despliegue y DevOps

1. **Contenedores y Orquestación:**
   - Docker: Creación de contenedores Docker para microservicios.
   - Docker Compose: Orquestación de múltiples contenedores para un entorno local.
   - Kubernetes: Introducción a la orquestación de contenedores, creación de pods, servicios, y despliegues.
2. **CI/CD (Integración y Despliegue Continuo):**
   - Herramientas de CI/CD como Jenkins, GitLab CI, GitHub Actions.
   - Creación de pipelines de CI/CD para automatizar el despliegue de microservicios.
3. **Infraestructura como Código:**
   - Herramientas de IaC como Terraform o AWS CloudFormation.
   - Gestión de infraestructura en la nube (AWS, Azure, GCP).
4. **Observabilidad y Monitoreo:**
   - Monitoreo de aplicaciones con herramientas como Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana).
   - Centralización de logs con ELK o herramientas como Splunk.
   - Health checks y métricas con Spring Boot Actuator.

Fase 6: Seguridad

1. **Autenticación y Autorización Avanzada:**
    - Implementación de OAuth2 y OpenID Connect (OIDC) para la autenticación de microservicios.
    - Implementación de seguridad basada en roles y permisos.
2. **Seguridad de API:**
    - Protecciones contra ataques comunes (CSRF, XSS, SQL Injection).
    - Implementación de políticas de CORS, HSTS y otras cabeceras de seguridad.
3. **Pruebas de Seguridad:**
    - Realización de pruebas de penetración.
    - Uso de herramientas de análisis de vulnerabilidades como OWASP ZAP o Snyk.

Fase 7: Testing y Calidad de Software

1. **Pruebas Automatizadas:**
    - Pruebas unitarias con JUnit y Mockito.
    - Pruebas de integración y end-to-end.
    - Pruebas de contrato para microservicios.
2. **Pruebas de Carga y Rendimiento:**
    - Pruebas de carga con herramientas como Apache JMeter, Gatling.
    - Monitoreo de rendimiento y optimización.
3. **Técnicas de Refactoring y Código Limpio:**
    - Principios SOLID, DRY (Don't Repeat Yourself), KISS (Keep It Simple, Stupid).
    - Técnicas de refactorización para mantener el código limpio y sostenible.

Fase 8: Gestión de Configuración y Entornos

1. **Gestión de Configuración:**
    - Uso de Spring Cloud Config o HashiCorp Vault para la gestión centralizada de configuración.
    - Encriptación de configuraciones sensibles y secretos.
2. **Manejo de Entornos:**
    - Configuración de múltiples entornos (desarrollo, pruebas, producción).

- Uso de perfiles de Spring Boot para configurar comportamientos específicos por entorno.

Fase 9: Diseño y Documentación de API

1. **Documentación de API:**
   - Uso de OpenAPI/Swagger para documentar APIs REST.
   - Generación automática de documentación y uso de herramientas como Springdoc OpenAPI.
2. **Gestión de Versiones de API:**
   - Estrategias de versionado de API (versionado en URL, headers).
   - Deprecación controlada y mantenimiento de compatibilidad.

Fase 10: Observabilidad y Monitoreo en Profundidad

1. **Monitoreo de Microservicios:**
   - Implementación de observabilidad distribuida: Tracing, logging, métricas.
   - Uso de OpenTelemetry o Zipkin para trazas distribuidas.
2. **Asegurar Disponibilidad y Escalabilidad:**
   - Configuración de alertas de monitoreo con Prometheus y Grafana.
   - Pruebas de escalabilidad y ajuste de parámetros de rendimiento.

Diagrama del roadmap para aprender Spring Boot: el objetivo es proporcionar una visión clara de las tecnologías esenciales, las recomendaciones más efectivas y los temas avanzados necesarios para dominar el stack tecnológico con Spring Boot y la programación en Java.

Recursos para Aprender

1. **Documentación Oficial:**
    - Spring Framework: spring.io
    - Java: Oracle Java Documentation
2. **Libros:**
    - "Spring Microservices in Action" de John Carnell.
    - "Designing Data-Intensive Applications" de Martin Kleppmann.
    - "Java Concurrency in Practice" de Brian Goetz.
    - "Think Java" de Allen B. Downey y Chris Mayfield.
    - "Kotlin Cookbook" de Ken Kousen.
    - "Java 8 Lambdas" de Richard Warburton.
    - "Stratospheric - From Zero to Production with Spring Boot and AWS" de Björn Wilmsmann, Philip Riecks y Tom Hombergs.
    - Spring Annotations Cheatsheet
3. **Cursos y Tutoriales:**
    - Codigo Facilito, Pluralsight, Coursera, Udemy, Spring Academy.
    - Youtube: La Tecnología Avanza, MitCode, CodelyTV, Java Brains, Amigoscode, freeCodeCamp.org
4. **Comunidades:**
    - Stack Overflow, Telegram, Facebook, Reddit, Gitter, Discord, Slack (Grupos de Java y Spring).
5. **Algunos recursos interesantes:**
    - Guía de desarrollo de código estilo Google
    - Resumen en español del libro Clean Code
    - Algunos Patrones de diseño