a) **Plan de Estudio**

Tema I: Introducción y conceptos de Java

- Paradigmas de programación
- Compilador
    - Tipo de compilación
        - AOT
        - JIT
    - Validación de la sintaxis
    - Orden de operaciones
    - Compilación del código
- Proyecto para aplicaciones Java
    - Maven
    - Gradle
- Manejo de dependencias
- Diferencias del JDK

Tema 2: Optimización del código

- Visualización del código compilado
- Diferencia del código escrito vs compilado
- Herramientas de monitoreo y perfil
    - Flight Recorder
    - VisualVM
- Depuración del código
- Manejo de logs

Tema 3: Pruebas con JUnit

- Concepto de prueba
- Tipos de pruebas
    - Prueba Unitaria
    - Prueba de Cobertura
    - Prueba de UI
- Uso de JUnit 5
- Integración de JUnit 5
- Ciclo de vida
- Integración con frameworks
- Decoradores o Annotations
    - @Test
    - @DisplayName
    - @BeforeEach
    - @AfterEach

        - @BeforeAll
        - @AfterAll
        - @Nested
        - @Tag
        - @Order
        - @TestMethodOrder(OrderAnnotation.class)

Tema 4: Orientación del código a las pruebas

- Construir y reconocer código testeable
- Características de una prueba
    - Automatizable
    - Completas
    - Repetibles
    - Independientes
  - Codificación orientada a pruebas
    - Principio de responsabilidad
    - Principio de repetibilidad
    - Parametrización
    - Retorno de resultados
- Creación de conjuntos de datos de prueba
- Escritura de pruebas
- Documentación en las pruebas
    - ¿Qué se debe comentar?
    - Manejo de recursos anexos en las pruebas
      - Gráficas
      - Documentos explicativos de un bug
      - etc
- Ejecución de las pruebas
- Mantenimiento de las pruebas

Tema 5: Seguridad

- Consideraciones en la codificación
- Vulnerabilidades más comunes OWASP Top 10
- Herramientas de escaneo de vulnerabilidades
    - Static Application Security Testing (SAST) Tools: codacy, sonarqube, jfrog xray, Veracode, Checkmarkx
    - Static Code Quality Tools
    - Dynamic Application Security Testing (DAST) Tools
    - Interactive Application Security Testing (IAST)
    - Dependency Scanner

- Prácticas que incurren en vulnerabilidades de seguridad
    - Ausencia de librerías
    - Desconocimiento del framework
    - Reconstruir código ya probado
    - Incursión en código viejo
    - Negación a la actualización de las herramientas
- ¿Cómo mejorar en el manejo de las vulnerabilidades?
- Mitigar vulnerabilidades más comunes

Tema 6: Diseño y desarrollo arquitectónico

- ¿Qué es un diseño arquitectónico?
- Diseño Monolítico
    - Reto de implementación
    - Ventajas competitivas
    - Desventajas
    - Problemas de diseño
    - Masificación del servicio
- Diseño en micro servicios
    - Reto de implementación
    - Ventajas competitivas
    - Desventajas
    - Problemas de diseño
    - Masificación del servicio
- Elementos a considerar para elegir una arquitectura

Ranking de las vulnerabilidades más frecuentes
https://owasp.org/www-project-top-ten/

Artículo OWASP sobre herramientas de escaneo de vulnerabilidades
https://owasp.org/www-community/Free_for_Open_Source_Application_Security_Tools

Conceptos de los diversos tipos de pruebas
https://es.wikipedia.org/wiki/Pruebas_de_software

Documentación oficial de JUnit 5
https://junit.org/junit5/docs/current/user-guide/

Artículo sobre Flight Recorder
https://www.baeldung.com/java-flight-recorder-monitoring

Página oficial de VisualVM
https://visualvm.github.io/

Artículo sobre los Profilers
https://www.baeldung.com/java-profilers

Página oficial de JFrog XRay
https://jfrog.com/xray/

Página oficial de Codacy
https://www.codacy.com/

Página oficial de Sonarqube
https://www.sonarqube.org/features/security/sast/