Desarrollo Full Stack con Kotlin: API, Mongo e IA Aplicada

Programa Oficial – 10 Sesiones

Semana 1 – Arquitectura y Plantilla Ktor

Descripción:

Se establecen las bases del backend y se comprende la arquitectura general del sistema.

Contenido:

- Arquitectura cliente-servidor
- Generación del proyecto desde start.ktor.io
- Estructura del proyecto
- Endpoint GET /health
- Pruebas con Postman

Herramientas necesarias:

- IntelliJ IDEA CE
- JDK 17+
- GitHub
- Postman
- Internet estable

Semana 2 – MongoDB y Guardado

Descripción:

El backend comienza a persistir datos reales en MongoDB Atlas.

Contenido:

- Conexión a MongoDB Atlas
- Modelo Garment
- Repository básico
- POST /garments

Herramientas necesarias:

- MongoDB Atlas
- MongoDB Compass
- IntelliJ IDEA
- Postman

Semana 3 – CRUD Completo

Descripción:

Se consolida la gestión completa de datos en la API.

Contenido:

- GET /garments
- GET /garments/{id}
- DELETE /garments/{id}
- Códigos HTTP y manejo de errores

Herramientas necesarias:

- MongoDB Atlas activo
- Postman
- IntelliJ IDEA

Semana 4 – Seguridad Básica (Basic Auth)

Descripción:

Se protege la API antes de exponerla públicamente.

Contenido:

- Implementación de Basic Auth en Ktor
- Variables de entorno
- Protección de endpoints
- Pruebas con Postman

Herramientas necesarias:

- IntelliJ IDEA
- Postman
- JDK 17+
- Internet estable

Semana 5 – Integración con OpenAI (Extract)

Descripción:

Se integra inteligencia artificial para extraer atributos desde imágenes.

Contenido:

- Cliente HTTP en Kotlin
- POST /garments/extract (multipart)
- Tipado de respuesta
- Pruebas con imagen en Postman

Herramientas necesarias:

- Cuenta OpenAI
- Postman
- IntelliJ IDEA
- Internet estable

Semana 6 – Integración + Motor de Sugerencias

Descripción:

Se conectan los componentes y se implementa el sistema de ranking determinístico.

Contenido:

- Flujo Extract → Save
- Ajustes del modelo Garment
- Reglas de combinación
- POST /outfits/suggest
- Sistema de scoring

Herramientas necesarias:

- MongoDB Atlas con datos
- Postman
- IntelliJ IDEA

Semana 7 – Docker + Deploy + HTTPS

Descripción:

El backend se prepara para producción y se despliega públicamente.

Contenido:

- Creación de Dockerfile
- Build de imagen
- Creación de droplet en DigitalOcean
- Configuración de dominio y HTTPS
- Pruebas públicas de la API

Herramientas necesarias:

- Docker Desktop
- Cuenta DigitalOcean
- Dominio
- MongoDB Atlas
- Internet estable

Semana 8 – Android Nativo Parte 1

Descripción:

Se crea la aplicación Android base y se conecta al endpoint health.

Contenido:

- Creación del proyecto Android
- Botón simple y recursos básicos
- Configuración Retrofit
- Llamada a GET /health
- Mostrar resultado en pantalla

Herramientas necesarias:

- Android Studio
- Emulador o dispositivo físico
- API pública activa
- Internet estable

Semana 9 – Android Nativo Parte 2

Descripción:

Se implementa el listado y visualización de datos reales desde la API.

Contenido:

- Modelo Kotlin equivalente a Garment
- GET /garments
- RecyclerView
- Estados UI (loading/error)
- Visualización de imágenes

Herramientas necesarias:

- Android Studio
- Emulador o dispositivo físico
- API pública activa
- Internet estable

Semana 10 – Android Nativo Parte 3

Descripción:

Se completa el flujo end-to-end con captura de imagen y sugerencias.

Contenido:

- Captura de imagen
- Multipart a /garments/extract
- POST /garments
- Formulario Outfit de hoy
- POST /outfits/suggest
- Demo final

Herramientas necesarias:

- Android Studio
- Emulador o dispositivo físico con cámara
- API pública HTTPS
- Internet estable