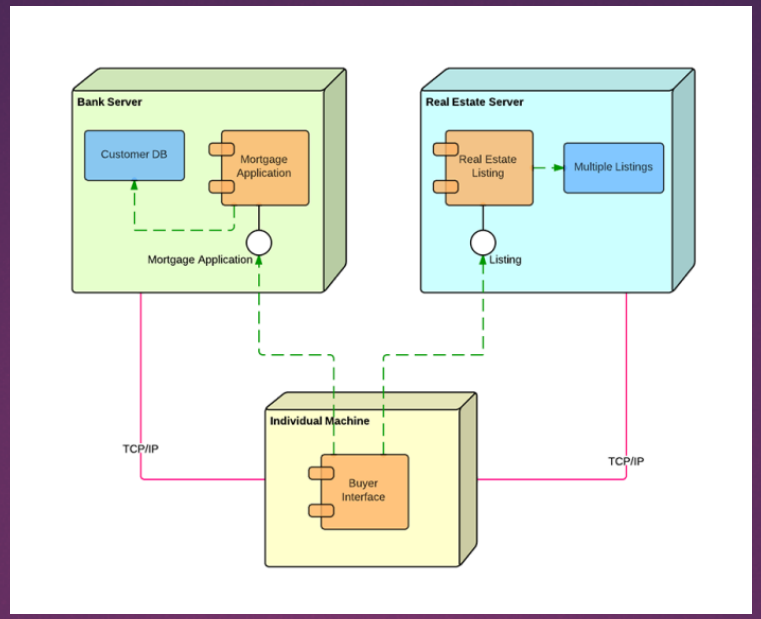


Diagrama de Despliegue

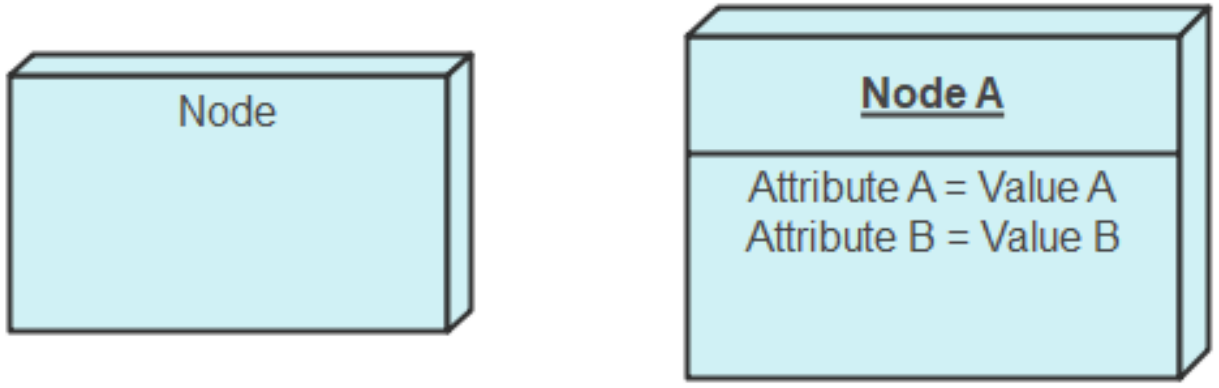
INSTRUCTORA ISAURA SUAREZ

Que es?

- Es un diagrama UML que muestra la arquitectura de ejecución

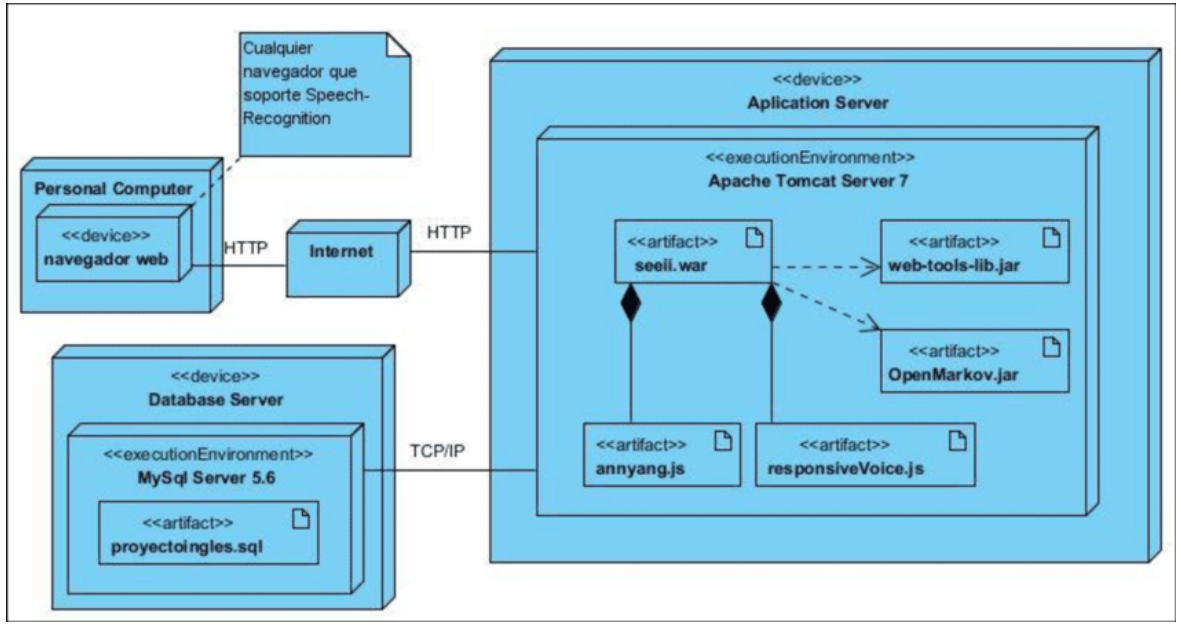
Elementos básicos

- Nodo (Node): entidad computacional (máquina física, VM, contenedor, dispositivo IoT). En UML se representa como un cubo o caja etiquetada.



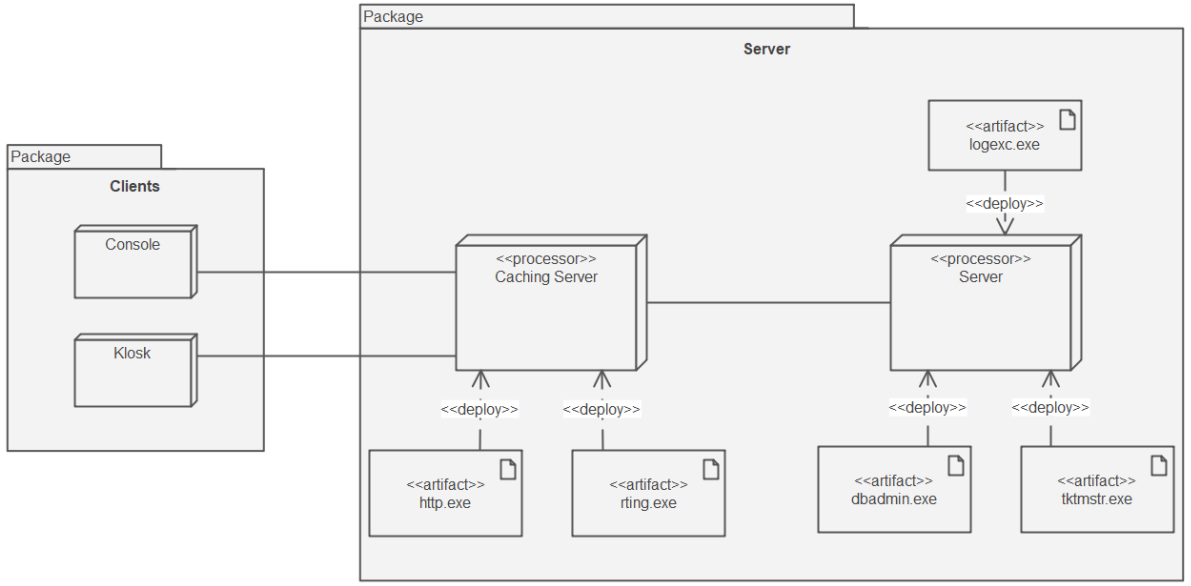
Elementos básicos

- ExecutionEnvironment / Software node: entorno de ejecución (JVM, Docker runtime, K8s node).



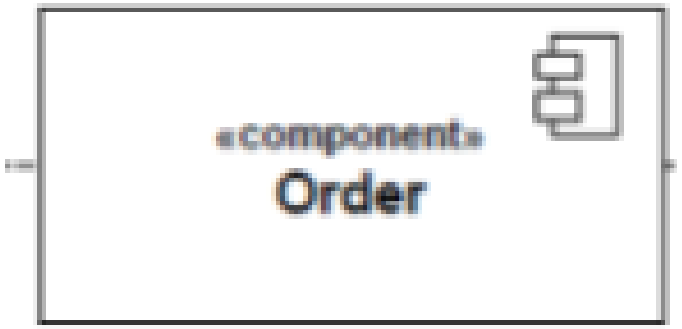
Elementos básicos

- Artifact (Artefacto): archivo o paquete desplegable (WAR, JAR, .zip, imagen Docker, script SQL).



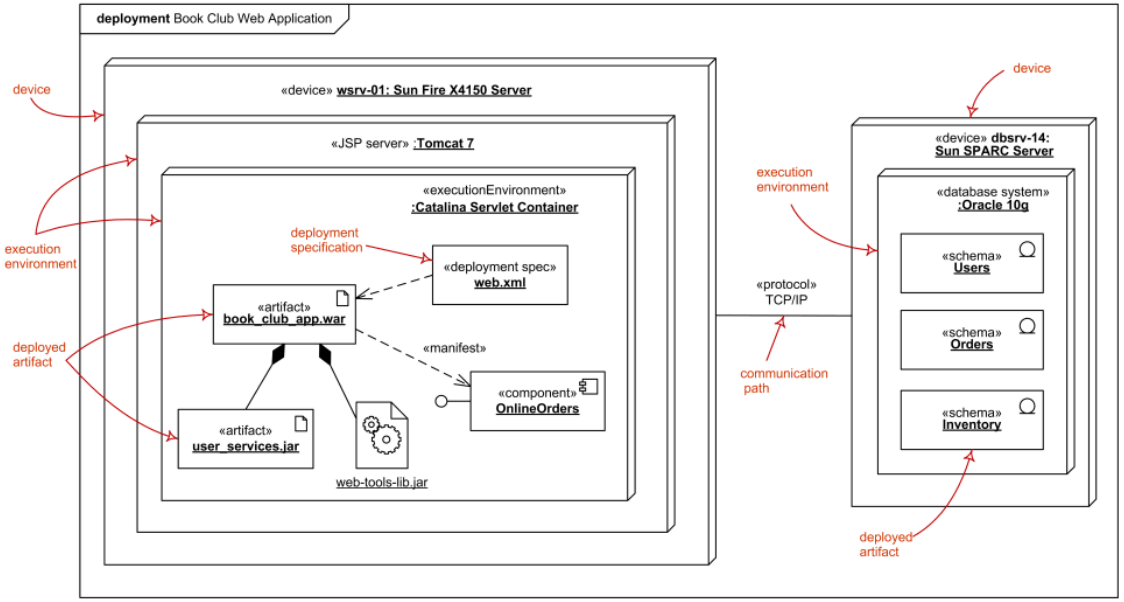
Elementos básicos

- Component (opcional): módulo lógico que vive dentro de un artefacto.



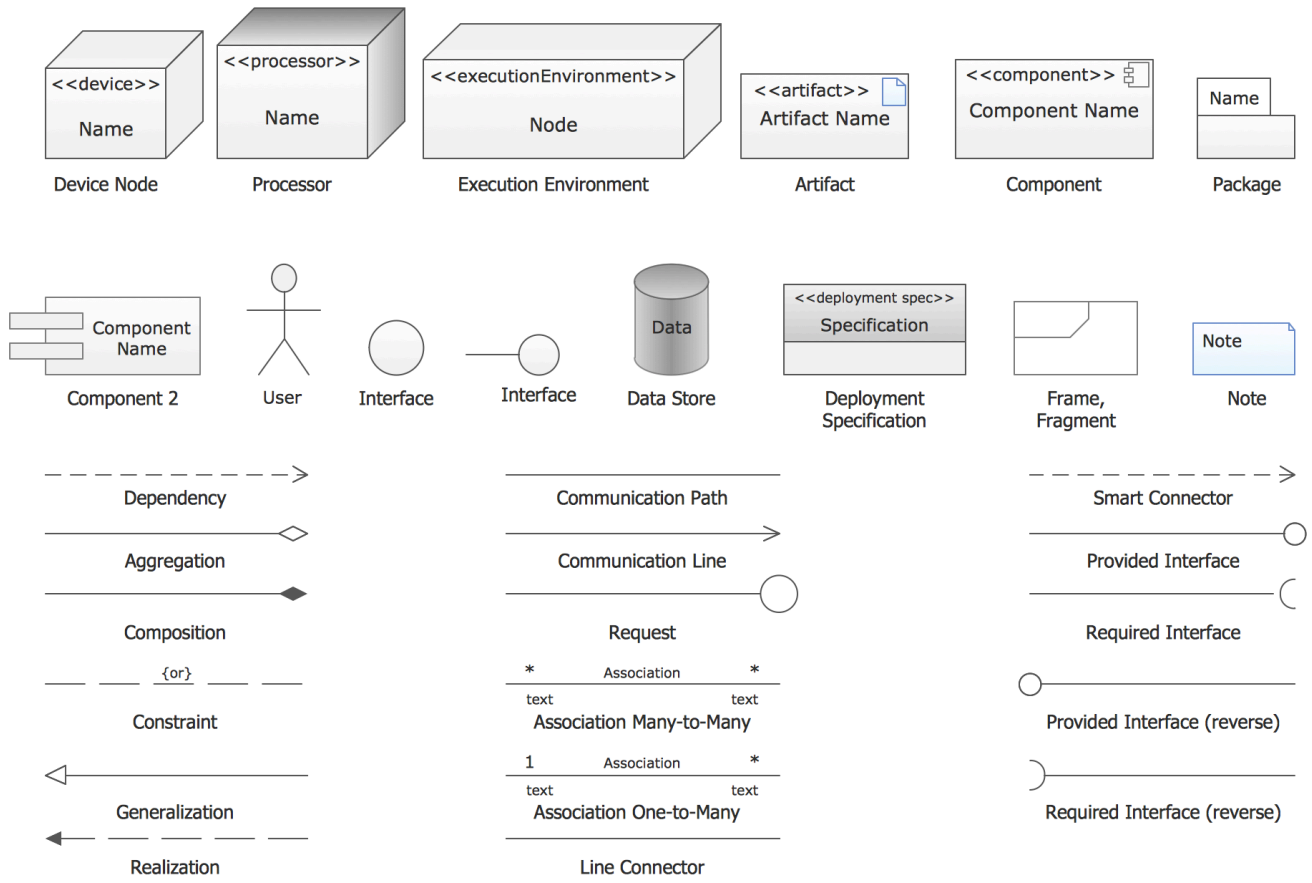
Elementos básicos

- Communication Path: línea entre nodos que indica canales/protocolos (HTTP, TCP, gRPC).



Elementos básicos

- DeploymentSpecification: parámetros de despliegue (variables env, tamaño, réplicas, CPU/RAM).



Herramientas recomendadas para crearlos

- diagrams.net / draw.io — interfaz gráfica, plantillas UML, fácil para estudiantes. draw.io
- PlantUML (texto) ⇒ diagrama (ideal para versionado, reproducible en repositorios). PlantUML.com
- Modeladores y tutoriales: Visual Paradigm, Creately, y otros (útiles para ejemplos y plantillas).

Paso a paso para elaborar un diagrama de despliegue

- Paso 0: Entorno del ejercicio (antes de dibujar)

  - Aclaren el alcance: ¿solo arquitectura local? ¿Cloud? ¿Microservicios o monolito?.

  - Definan requisitos no funcionales: disponibilidad, tolerancia a fallos, número de usuarios concurrentes.

- Paso 1: Inventario de artefactos Listado mínimo:

  - Frontend (ej. SPA React → artefacto frontend.zip o imagen frontend:1.0).

  - Backend (API — api.jar o imagen Docker).

  - Base de datos (MySQL, PostgreSQL), scripts de esquema.

  - Caché (Redis), cola (RabbitMQ), otros servicios externos (SMTP, storage S3).

  - Agentes de monitoring / logging (Prometheus, Fluentd).

Paso a paso para elaborar un diagrama de despliegue

- Paso 2: Identificar nodos físicos/lógicos
  - Cliente (navegador móvil/desktop).
  - Gateway / Load Balancer (Nginx, ELB).
  - Servidores de aplicaciones (VMs, contenedores, pods).
  - Servidor(s) de BD (cluster/replica).
  - Servicios externos / proveedores (correo, pagos).

Nota: Agrupar en subredes / zonas (público vs privado) si aplica.

- Paso 3: Mapear artefactos sobre nodos
  - Para cada artefacto, anotar en qué nodo corre y bajo qué entorno de ejecución (OS, versión de Java/Python, runtime de Node, Docker/Kubernetes). Añadir puertos y protocolos (ej. HTTP 443, TCP 3306).

Paso a paso para elaborar un diagrama de despliegue

- Paso 4: Dibujar comunicaciones y dependencias

  - Añadir líneas entre nodos con etiqueta de protocolo/puerto y latencia esperada si se conoce.

  - Indicar direcciones públicas, NAT, firewalls/security groups y reglas principales.

- Paso 5: Anotar especificaciones de despliegue

  - Réplicas / auto-scaling / tamaño de instancia (vCPU, RAM, disco).

  - Políticas de backup y réplica (por ejemplo: MySQL master-replica, backup diario).

  - Ruta de logs, retención, y herramienta de observabilidad.

Paso a paso para elaborar un diagrama de despliegue

- Paso 6: Seguridad y zonas

  - Marcar límites de red (DMZ, subred privada).

  - Indicar certificados TLS, autenticación (JWT / OAuth), y puertos expuestos.

- Paso 7: Revisión y versionado

  - Añadir fecha, autor, versión del diagrama.

  - Subir al repositorio del proyecto (README con enlace al diagrama o incluir PlantUML en docs/ para renderizar).

Ejemplo práctico

```
@startuml ' Diagrama de despliegue simple: Frontend, LB, App (Docker), DB (MySQL), Redis cloud "Internet" as Internet

node "Cliente\n(Navegador)" as Client

node "Load Balancer\n(Nginx / ELB)" as LB

node "App Host\n(Docker / K8s Node)" as AppHost {

artifact "app-image:1.0" as AppImage

}

node "DB Host\n(MySQL Primary)" as DBHost {

artifact "mysql-data" as MySQL

}

node "Cache\n(Redis)" as RedisHost {
artifact "redis-data" as Redis

}

Client --> LB : HTTPS 443 LB --> AppHost : HTTP 80 AppHost --> DBHost : TCP 3306 AppHost --> RedisHost : TCP 6379
@enduml
```

Recursos y lecturas

Les dejo enlaces útiles para el material de apoyo, cada uno con ejemplos y plantillas:

- **Visual Paradigm. Qué es un Deployment Diagram (tutorial).** definición, ejemplos y pasos. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/ visual-paradigm.com
- **UML-Diagrams.org. Deployment diagrams (notación y ejemplos)** referencia de notación. https://www.uml-diagrams.org/deployment-diagrams.html uml-diagrams.org
- **PlantUML. página oficial** (ejemplos y sintaxis para deployment diagrams). https://plantuml.com/deployment-diagram (o https://plantuml.com) PlantUML.com
- **Cómo crear diagramas de despliegue con draw.io (tutorial práctico)** paso a paso con plantillas. https://drawio-app.com/blog/create-uml-deployment-diagrams-in-draw-io/ draw.io
- **Kubernetes. conceptos de arquitectura** si el proyecto usa K8s, entender cluster/nodes/pods/namespaces. https://kubernetes.io/docs/concepts/architecture/