

# Checklist para el Obligatorio

El checklist está organizado por secciones con el puntaje correspondiente e incluye todos los requisitos específicos que debe cumplir el obligatorio, la entrega y la defensa.

## Entrega General

- ☐ **Registro de equipo completado** en el sistema de gestión
  - ☐ **Archivo .rar o .zip** preparado con todo el contenido
  - ☐ **Defensa programada** (presentación de 20 min máximo)
- 

## 1. Planificación y Seguimiento (5 pts)

### Tablero Kanban

- ☐ Tablero Kanban configurado y en uso
  - ☐ Tareas claramente definidas en el tablero
  - ☐ **Screenshots documentados** del flujo de trabajo en **3 momentos distintos**
  - ☐ Evidencia de progreso y seguimiento consistente
- 

## 2. Integración y Entrega Continua - CI/CD (4 pts)

### Pipelines

- ☐ Pipeline para ambiente **DEV** implementado
- ☐ Pipeline para ambiente **TEST** implementado
- ☐ Pipeline para ambiente **PROD** implementado
- ☐ Cada pipeline tiene **mínimo 2 etapas**
- ☐ **Quality gates** implementados entre ambientes

### Validación

- ☐ Pipelines se ejecutan correctamente
- ☐ Quality gates funcionan como barrera de calidad

---

### 3. Containerización y Despliegue (5 pts)

#### Containers

- ☐ Dockerfiles optimizados y funcionando
- ☐ Images construidas correctamente
- ☐ Buenas prácticas aplicadas

#### Orquestación

- ☐ Despliegue en **AWS ECS/EKS** (o equivalente en otra nube)
  - ☐ Servicios de orquestación funcionando correctamente
  - ☐ Aplicación accesible desde internet
- 

### 4. Testing y Calidad (4 pts cada uno)

#### Testing (4 pts)

Implementar **UNO** de los siguientes:

- ☐ **Pruebas funcionales** (Postman, Selenium, etc.)
- ☐ **Pruebas de carga/rendimiento** (JMeter, k6, etc.)
- ☐ Testing integrado en el pipeline CI/CD

#### Análisis de Código Estático (4 pts)

- ☐ Herramienta integrada (SonarQube, ESLint, etc.)
  - ☐ Análisis ejecutándose en el pipeline
  - ☐ **Informe detallado** con hallazgos significativos
  - ☐ **Recomendaciones de mejora** documentadas
-



## 5. Control de Versiones (5 pts + 4 pts)

### Repositorios Git (5 pts)

- ☐ Todo el código almacenado en repositorios Git
- ☐ Repositorio/s bien organizados
- ☐ Separación adecuada de componentes

### Estrategias de Ramificación (4 pts)

- ☐ **Código de aplicación:** Git Flow O Trunk Based implementado
  - ☐ **Justificación** de la elección documentada
  - ☐ **Código de infraestructura:** Feature Branch implementado
  - ☐ **Pull Requests** con revisiones entre miembros del equipo
  - ☐ Evidencia clara de **trabajo colaborativo**
- 



## 6. Infraestructura como Código - IaC (4 pts)

### Terraform

- ☐ Toda la infraestructura implementada con Terraform
  - ☐ **Modularización** implementada (opcional pero recomendada)
  - ☐ **Variables parametrizadas** por ambiente
  - ☐ **Outputs relevantes** documentados
  - ☐ **Manejo seguro de secretos** implementado
  - ☐ Código IaC funcional y desplegable
- 



## 7. Servicios Serverless (4 pts)

### Implementar AL MENOS UNO:

- ☐ **AWS Lambda** para automatizaciones (alertas, backups, análisis de logs, etc.)
- ☐ **API Gateway** para gestión de acceso y seguridad a APIs
- ☐ Servicio con **propósito claramente definido**
- ☐ **Integración** con el resto de la arquitectura
- ☐ Funcionamiento correcto del servicio



## 8. Observabilidad (4 pts)

### Monitoreo

- ☐ Solución básica de **monitoreo** implementada
  - ☐ **1 dashboard** mínimo para métricas críticas
  - ☐ **2 alertas** mínimo para condiciones relevantes
  - ☐ Herramientas de observabilidad funcionando
- 



## 9. Documentación (5 pts)

### README.md

- ☐ **README.md completo** con instrucciones de despliegue
- ☐ Instrucciones de **configuración** detalladas
- ☐ Documentación clara y bien estructurada

### Diagramas

- ☐ **Diagramas de arquitectura** creados
- ☐ **Flujos CI/CD** diagramados
- ☐ **Estrategia de Git** documentada

### Decisiones de Diseño

- ☐ **Decisiones de diseño** importantes documentadas
  - ☐ **Lecciones aprendidas** documentadas
  - ☐ Documentación en **formato Markdown**
- 



## 10. Implementación General (7 pts)

### Solución Completa

- ☐ La solución resuelve el **problema propuesto** completamente
- ☐ Implementación **innovadora**
- ☐ Demuestra **entendimiento de DevOps**
- ☐ Aplicación Voting-app desplegada y funcionando
- ☐ Todos los componentes integrados correctamente

---

## 11. Defensa

### Preparación

- ☐ **Presentación preparada** (máximo 20 minutos)
  - ☐ Enfoque en **presentación de venta** (no técnica)
  - ☐ **Cámara y micrófono** listos para la defensa
  - ☐ Todos los miembros preparados para participar
- 

## 12. Entrega Final

### Contenido del Archivo

- ☐ **Repositorio Git** completo incluido
  - ☐ **Documentación** completa incluida
  - ☐ **Imágenes/Screenshots** incluidas
  - ☐ **Archivo .rar o .zip** correctamente empaquetado
  - ☐ Todos los artefactos necesarios incluidos
- 

✨ ¡Revisen cada punto antes de la entrega final! ✨