

## Entregable 4 - Integración y deployment continuo, refactoring y code smells

**Fecha límite** de entrega: 18/11 por tarea de Moodle.

Grupo de 2 personas como máximo: Avisar al docente en clase si cambia la conformación del grupo.

Fecha de presentación en clase: 18/11 - Deben estar los dos integrantes presentes.

**Forma de entrega:** Se debe subir un zip, con el nombre Entregable2Apellido1Apellido2.zip, ejemplo (Entregable2GarciaPerez.zip) que contenga los apellidos de los integrantes del grupo.

El zip debe contener:

1. Código fuente y archivos de configuración del proyecto completo, de manera similar a los ejemplos vistos en clase (tests, Jenkinsfile, pom.xml, dependencies.txt etc etc)
2. Documentación técnica de tipo informe PDF. Debe incluir obligatoriamente:
  - a. Decisiones técnicas tomadas
  - b. Estructura del proyecto
  - c. Pipelines utilizados y una breve justificación de cada paso del mismo
  - d. Indicar el code smell trabajado, la técnica de refactoring elegida para solucionarlo e indicar el código que se cambia.
  - e. Aprendizajes obtenidos de este tema
  - f. Breve conclusión final sobre el curso, tema más interesante y qué sugeriría para mejorar

# Requerimientos del proyecto Mi Playlist:

## Parte 1: Desarrollo

Se desea crear una pequeña webapp con Java que maneje una playlist de videos de un tema de tu elección (música, películas, videojuegos, deportes).

La webapp debe cumplir con estos requerimientos, en orden de prioridad:

1. El usuario puede agregar y quitar videos con nombre y link en esta lista.
2. Permitir visualizar estos videos de manera embebida en nuestra web.
3. Persistir esta lista de la manera que mejor consideren, de modo que entre una ejecución y otra se mantengan los cambios realizados
4. Utilizar alguna herramienta que consideren adecuada para que la **UI sea atractiva** para un usuario final.
5. Sumar likes a los videos
6. Marcar/desmarcar los videos favoritos.

## Parte 2: CI / CD

Necesitamos automatizar la integración y el despliegue a producción.

Los requerimientos de esta parte son:

1. Utilizar Git o la herramienta que prefieran para hacer control de versiones de todos los archivos de este proyecto
2. Configurar Jenkins de manera local, con un pipeline (o más) que automatice estos pasos:
  - a. Tome el código del repositorio
  - b. Haga el build de la aplicación
  - c. Corra tests automáticos (sugerencia, puede usar JUnit)
  - d. Realice el deploy
3. Entregar scripts de deployment tanto para Mac como para Windows

## Sugerencias:

- Trabaje de manera incremental. Comience por crear una web sencilla sin todo lo demás, un “Hola Mundo” y realice toda la parte de CI/CD para conocer estas herramientas. Luego integre las demás funcionalidades de a poco.
- Cumpliendo con los requerimientos, se pueden agregar otras herramientas que prefieran o consideren adecuadas para lograr lo solicitado

## Presentación en clase:

1. Mostrar la aplicación web desarrollada.
2. Realizar un cambio en el código fuente, debe arreglar un code smell utilizando una técnica de refactoring. Realizar la integración y despliegue automático con Jenkins.
3. Explicar brevemente cómo se implementaron los requerimientos de la parte 1 y cómo se construyó el/los pipeline/s de la parte 2.