

# Introducción a la Ingeniería del Software.

## Rúbricas de entregas de proyecto.

### E2. Propuesta Proyecto y Plan de Riesgo

- 30%. Problema explicado adecuadamente
- 10%. Roles de los integrantes son adecuados
- 10%. Hay al menos 5 riesgos especificados adecuadamente
- 10%. Los riesgos tienen tipo (ámbito) adecuado
- 10%. Los riesgos tienen descripción adecuada
- 10%. Los riesgos tienen probabilidad adecuada
- 10%. Los riesgos tienen efectos adecuados
- 10%. Los riesgos tienen estrategia adecuada

### E3. Planificación

- 20%. Modelo de proceso de software
- 20%. Modelo de proceso de software. Por qué
- 20%. Lista de tareas
- 20%. Secuenciación de tareas
- 20%. Explicación de las tareas

### E4. Requisitos

- 10%. Diagrama. Hay al menos tantos RFs adecuados como 5x miembros
- 10%. Diagrama. Hay al menos tantos RNFs adecuados como 1x miembros
- 10%. Diagrama. Las relaciones son adecuadas
- 10%. Historias de usuario. Hay al menos tantos RFs adecuadas como 5x miembros
- 10%. Historias de usuario. Hay al menos tantos RNFs adecuadas como 1x miembros
- 10%. Historias de usuario. Los "como" son adecuados.
- 10%. Historias de usuario. Los "quiero" son adecuados
- 10%. Historias de usuario. Los "para" son adecuados
- 10%. Pruebas de aceptación. Son adecuadas.
- 10%. La lista de requisitos parece reflejar una aplicación completa

### E5. Casos de Uso

- 10%. Diagrama. Hay al menos tantos casos de uso adecuados como 5x miembros.
- 10%. Diagrama. Los actores son adecuados.
- 10%. Diagrama. Las relaciones son adecuadas.

- 10%. Descripciones. Los títulos son adecuados
- 10%. Descripciones. Los contextos de uso son adecuados
- 10%. Descripciones. Las precondiciones son adecuadas
- 10%. Descripciones. Las postcondiciones son adecuadas
- 10%. Descripciones. Los escenarios principales son adecuadas
- 10%. Descripciones. Los escenarios alternativos son adecuadas
- 10%. Los casos de uso reflejan completamente los requisitos.

## E6. Modelo de dominio

- 10%. Clases. Hay al menos tantas clases adecuadas como 2x miembros.
- 10%. Clases. Los nombres son adecuados
- 10%. Clases. Los atributos son adecuados
- 10%. Relaciones. Las relaciones entre clases son adecuadas
- 10%. Relaciones. Los nombres de las relaciones son adecuados
- 10%. Relaciones. Los tipos de las relaciones son adecuados
- 10%. Relaciones. La dirección de las relaciones son adecuados
- 10%. Relaciones. La cardinalidad de las relaciones son adecuados
- 20%. El conjunto de clases y relaciones refleja todos los casos de uso

## E7. Diagramas de secuencia

- 20%. Diagramas. Hay tantos diagramas de secuencia adecuados como 2x el número de miembros.
- 10%. Diagramas. Cada diagrama se acompaña de un texto explicativo adecuado
- 10%. Líneas de vida. El uso de líneas de vida es adecuado
- 10%. Mensajes. El envío de mensajes es adecuado
- 10%. Mensajes. La recepción de mensajes es adecuada
- 10%. Mensajes. La secuenciación de mensajes es adecuada
- 10%. Mensajes. El uso de mensajes síncronos y asíncronos es adecuado
- 10%. Fragmentos. El uso de alt es adecuado
- 10%. Fragmentos. El uso de loop es adecuado

## E8. Presentación

- 10%. Transparencias. La "Descripción Del Problema" describe el problema en suficiente detalle.
- 10%. Transparencias. Los "Casos De Uso" reflejan el principal uso de la aplicación adecuadamente.
- 10%. Transparencias. El "Modelo De Dominio" refleja adecuadamente el dominio de la aplicación.
- 10%. Transparencias. La "Lecciones Aprendidas" son útiles y relevantes.
- 10%. Presentación. El problema se explica adecuadamente.

- 10%. Presentación. Los casos de uso se explican adecuadamente.
- 10%. Presentación. El modelo de dominio se explica adecuadamente.
- 10%. Presentación. Se demuestra adecuadamente el correcto funcionamiento de 2 casos de uso representativos de la aplicación.
- 10%. Presentación. Las lecciones aprendidas se explican adecuadamente.
- 10%. Presentación. Todos los integrantes del grupo presentan durante una cantidad de tiempo similar.

## E9. Implementación y pruebas

- 20%. Implementación. Hay tantos casos de uso implementados adecuadamente como número de miembros.
- 10%. Implementación. Todas las pruebas pasan porque la implementación es correcta.
- 10%. Implementación. La implementación sigue un estilo de arquitectura de software.
- 10%. Implementación. Se explica adecuadamente por qué se ha seleccionado ese estilo de arquitectura de software.
- 10%. Pruebas. Hay una prueba de integración adecuada para cada caso de uso implementado.
- 10%. Pruebas. Hay tantas pruebas unitarias adecuadas como 3x número de miembros.
- 10%. Pruebas. Las pruebas unitarias cubren tantas clases como 0.5x número de miembros.
- 10%. Pruebas. Se aporta un mecanismo adecuado para ejecutar todas las pruebas con un solo comando.
- 10%. Pruebas. Se usa un framework de pruebas unitarias.