

Proyecto de Curso

El proyecto de curso debe ser planteado por su grupo cumpliendo con las [Especificaciones del Proyecto Final](#). La entrega única de la dirección al repositorio debe hacerse antes de la primera entrega. A partir de la primera entrega y en adelante todas las entregas deben hacerse a través de GitHub, con un tag sobre el último commit de dicha entrega con la fecha y hora máxima estipulada en el enlace respectivo.

Para cada entrega, ya que todas deben estar en el mismo repositorio, no hay que entregar nuevamente la dirección del repositorio, en cambio si deben actualizar el readme.me del repositorio explicando el contenido del mismo, enlazando apropiadamente los documentos y explicando de qué se trata el proyecto (el nombre del proyecto ni su descripción debería hacer mención de que es un proyecto de curso, sino solo como un desarrollo orientado a resolver el problema planteado por ustedes).

El último commit de cada entrega debe ser etiquetado con el tag MilestoneX. De esta manera, podrán seguir haciendo commits, pero se sabrá sin ambigüedad hasta donde está el desarrollo de cada una.

Primera Entrega

En esta entrega debe subir el documento pdf al repositorio del proyecto en GitHub, y por tanto debe indicar la url del repositorio en la opción de la entrega donde se puede escribir texto. Su repositorio debe incluir un readme.md en el directorio principal donde se explique brevemente en qué consiste el proyecto, enlace a cada documento del proyecto y un listado de los integrantes con enlaces a sus usuarios de GitHub. El tag del último commit de esta entrega debe llamarse Milestone1.

Su entrega debe ser un documento en formato pdf con los siguientes elementos:

1. **Listado de integrantes** del grupo que trabajará en el proyecto.
2. **Enunciado del proyecto.** Debe redactar un enunciado en el cual explique con claridad en qué consiste el proyecto que propone desarrollar. Debe incluir un título o nombre para su proyecto.
3. **Requerimientos Funcionales.** Debe escribir la primera versión de los requerimientos funcionales de la totalidad del proyecto de tal forma que quede aún más claro que es lo que se espera desarrollar.
4. **Justificación.** Explicar de manera suficiente por qué el proyecto requiere la cantidad de integrantes indicados en el primer ítem.
5. **Sketches, Mockups ó Wireframes.** Pantallazos dibujados de las diferentes pantallas que tendrá su programa una vez esté terminado.
6. El **readme.md** del repositorio debe explicar brevemente (en inglés) de qué se trata el proyecto. Deben enlazar los archivos que documentan el proyecto (en formato pdf) y deben especificar las condiciones técnicas del mismo (lenguaje, sistema operativo y ambiente de desarrollo).

La rúbrica con la que será calificada esta entrega es la siguiente:

La entrega es solo un archivo en formato pdf y su nombre debe tener solo los primeros apellidos de los integrantes separados por guión.	Incluye el nombre completo de todos los participantes con su código estudiantil y un título apropiado para el proyecto.	La ortografía, vocabulario o gramática del enunciado es adecuada.	El enunciado presenta con claridad el problema y los elementos principales a abordar en el desarrollo del proyecto.	Los requerimientos funcionales no están en formato de tabla, están separados adecuadamente y su redacción es clara permitiendo conocer lo que se espera que el programa lleve a cabo.	La justificación permite conocer el alcance del proyecto y la dificultad inherente al desarrollo. Queda claro por qué se necesita la participación del número de integrantes indicada.	Mockups de la interfaz gráfica que expliquen cómo quedará su programa cuando esté terminado. Cada pantallazo o dibujo de la interfaz gráfica debe tener un texto que explique brevemente la funcionalidad posible en esa parte.
5%	10%	15%	20%	20%	10%	20%

Segunda Entrega

Su entrega debe incluir las dos siguientes partes:

- Un único documento en formato pdf con todos los elementos del análisis y el diseño que se indican a continuación:
 - Requerimientos Funcionales.
 - Requerimientos No Funcionales.
 - Diagrama de Clase del Modelo que incluya herencia y polimorfismo (clases abstractas, superclases no abstractas e interfaces).
 - Diagrama de Clase de la Interfaz con el Usuario.
 - Diseño de las Pruebas Unitarias.
 - Diagrama de Clases de las Pruebas Unitarias.
- El proyecto en eclipse es accesible de forma privada en GitHub (deben agregar como colaborador al monitor y al profesor), que incluya al menos 6 commits en días diferentes por cada integrante del grupo. En esta entrega, es necesario que el programa presente lo siguiente:
 - Interfaz gráfica implementada y completamente funcional en su navegación. Es decir, es posible ejecutar el programa y entrar a cada una de las opciones del programa aunque éstas aún no respondan a los requerimientos.
 - Implementación de al menos dos (2) requerimientos funcionales.
 - También se debe actualizar el **readme.md** del repositorio, en el cual se debe explicar brevemente (en inglés) de qué se trata el proyecto. Deben enlazar los archivos que documentan el proyecto (en formato pdf).

Importante definir un **tag** en Git para marcar el commit a revisar de esta entrega. El **tag** debe llamarse Milestone2.

Los requerimientos funcionales cubren completamente el proyecto a desarrollar, están redactados de forma clara y divididos	Los requerimientos funcionales enuncian correctamente las restricciones y condiciones no funcionales relacionadas	El diagrama de clases del modelo presenta un diseño apropiado que incluye herencia y polimorfismo. Se evidencia una adecuada distribución de responsabilidades	El diagrama de clases de la GUI presenta una adecuada distribución de	Diseña pruebas unitarias interesantes (verificando casos normales y extremos) para todos los métodos
--	---	--	---	---

apropiadamente	por ejemplo con GUI y Persistencia	entre clases. La notación es correcta.	responsabilidades y correcta notación.	no triviales (no getters, setters, ni constructores).
10%	5%	20%	5%	15%
El diagrama de clases de las pruebas unitarias tiene una clase de prueba por cada clase a probar. Los tipos de relaciones concuerdan con la forma como se maneja la clase en la prueba (asociación o dependencia). Tiene los métodos de configuración de escenarios y de pruebas.	El programa entregado se puede ejecutar. En la versión de entrega debe hacer una marca (tag en Git) donde indique que es la versión de la segunda del proyecto. Es posible acceder a todas las opciones del programa aunque todavía el programa esté completo.	Un requerimiento funcional está completamente implementado y se indica cuál es en el documento con el informe.	Otro requerimiento funcional está completamente implementado y se indica cuál es en el documento con el informe.	Entrega todos los artefactos del análisis y el diseño en un solo documento en formato pdf.
5%	15%	10%	10%	5%

Tercera Entrega

Su proyecto completamente funcional y cumpliendo con todos los requisitos. Se calificará utilizando la [Rúbrica de Proyecto de Curso](#), en sustentación cuya fecha será informada oportunamente por el profesor y a la cual tendrá que inscribirse para seleccionar un horario.

Importante definir un tag en GitHub para marcar el commit a revisar de esta entrega. El tag debe llamarse Milestone3.

La nota final del proyecto se calculará de la siguiente manera:

Entrega 1: 15%

Entrega 2: 25%

Entrega 3: 60%

Video explicativo de la entrega final. En el video deben presentar todos los integrantes del equipo. En el momento en que una persona se encuentre hablando debe visualizarse su video en una miniatura. En el video deben presentar de forma breve la documentación solicitada, haciendo énfasis en los elementos más importantes de esta entrega, al igual que una demostración de la implementación. Duración máxima del video: 5 minutos.