

Objetivo general del proyecto

El objetivo general de este proyecto es practicar varias etapas del desarrollo de una aplicación de software, desde el análisis hasta la construcción una aplicación funcional. A través del proyecto, los estudiantes pondrán en práctica todas las habilidades desarrolladas en el curso.

Objetivos específicos del proyecto

Durante el desarrollo de este proyecto se buscará el desarrollo de las siguientes habilidades:

1. Identificar y abstraer entidades en un dominio particular, así como sus características y relaciones.
2. Construir diagramas de clase UML para expresar modelos de dominio y modelos de diseño.
3. Diseñar una aplicación basada en el paradigma orientado a objetos y expresar el diseño, justificando las decisiones importantes que hayan tomado.
4. Implementar un diseño utilizando el lenguaje de programación Java.

Instrucciones generales

A continuación, encontrará la definición inicial del proyecto, que será la base del trabajo de los 3 proyectos que se construirán durante todo el semestre.

El trabajo del proyecto #2 debe partir de la base del proyecto #1, y el trabajo del proyecto #2 será la base del proyecto #3. Si es necesario, podrán hacerse cambios en cada una de las etapas para mejorar o corregir decisiones que se hayan tomado en las etapas anteriores.

En el proyecto 1 no es necesario que haya una interfaz gráfica: la implementación que realicen puede tener una interacción con el usuario basada en consola.

El proyecto debe desarrollarse en **grupos que deben mantenerse durante todo el semestre**.

Contexto del proyecto

Dentro del proyecto del curso vamos a construir una aplicación para hacerle seguimiento a un proyecto realizado por una o por varias personas.

Las siguientes son las principales características que el sistema tiene que soportar.

1. Cada proyecto debe tener un nombre, una descripción, una fecha de inicio y una fecha estimada de finalización si ya se conoce.
2. Al crear un proyecto debe haber un participante inicial (el dueño del proyecto), pero nuevos participantes pueden agregarse más adelante. No es necesario poder eliminar participantes. De cada participante debemos saber el nombre y un correo.

3. Cada participante puede registrar que realizó una actividad como parte del proyecto. Cada actividad tiene un título, una descripción, un tipo, una fecha en la que se realizó, una hora de inicio, una hora de finalización y un participante que la realizó.
4. Por defecto, la fecha y hora en la que se realice una actividad serán las del momento en que se esté haciendo el registro, pero debe ser posible modificar estos datos. Esto es útil para registrar, por ejemplo, tareas que se hicieron horas o días antes, mientras no se tenía acceso al sistema.
5. Puede haber varios registros de actividades con el mismo nombre. Eso serviría para indicar, por ejemplo, que hubo una tarea que se hizo durante varios días.
6. Por defecto la persona que está usando la aplicación y haciendo el registro de una actividad es el miembro del proyecto que la realizó. Sin embargo, debe ser posible registrar una actividad realizada por alguien más. Nota: No es necesario que el sistema tenga algún sistema de autenticación.
7. El tipo de una actividad sirve para clasificarlas: los tipos disponibles se deben definir para cada proyecto. Por ejemplo, los tipos en un proyecto podrían ser “Documentación”, “Implementación”, “Pruebas”, “Investigación”, “Diseño”, etc.
8. La aplicación debe permitir *cronometrar* una actividad. Es decir que el usuario puede indicar cuando vaya a empezar a hacer una actividad y cuando la haya terminado y el sistema debe calcular automáticamente el tiempo que tomó. Este cronómetro debe poderse pausar y continuar. No es necesario que exista la posibilidad de cronometrar varias tareas simultáneamente.
9. La aplicación debe ser capaz de mostrar un reporte de las actividades de cada miembro del equipo incluyendo información como: tiempo total invertido, tiempo promedio por tipo de actividad, tiempo por cada día, etc.

Aspectos técnicos

Toda la información debe ser persistente y se deben utilizar archivos planos (txt, csv, tsv, etc.). La idea es que estos archivos puedan almacenarse y sincronizarse en un repositorio GIT, así que usando archivos planos es más fácil que no haya problemas en el momento de sincronizar.

Usted puede definir la estructura de los archivos como quiera, pero debe haber un archivo por proyecto. Si es necesario, su aplicación podría utilizar archivos temporales adicionales, pero su aplicación debe funcionar correctamente aún si estos archivos se eliminan.

Si en un proyecto hay varios participantes, se esperaría que cada uno registre sus actividades en su propio computador: al momento de sincronizar los archivos de proyectos por medio de un repositorio GIT, la mezcla (*merge*) de los archivos de proyecto debería permitir construir una visión global del avance del proyecto.

La interfaz de la aplicación debe estar basada en consola.

No es necesario que se soporten funcionalidades que no hayan sido mencionadas, pero se pueden incluir funcionalidades adicionales si les facilita el trabajo.

Entrega 1: Análisis del proyecto

La primera etapa del proyecto consiste en realizar el análisis del sistema que debe construir.

Actividades

1. Construya un modelo de dominio (diagrama de clases) a partir de la información del caso:
 - a. Identifique las entidades que aparecen dentro del caso.
 - b. Identifique las características (atributos) de esas entidades.
 - c. Establezca las relaciones entre las entidades, incluyendo asociaciones y relaciones de herencia.

2. Construya un documento de análisis para su proyecto. El documento debe incluir:
 - a. El modelo de dominio que construyó en el paso anterior.
 - b. Una descripción de los requerimientos funcionales que su aplicación soportará. Le recomendamos utilizar historias de usuario.
 - c. Una descripción con las restricciones del proyecto.

Entrega

1. El proyecto debe entregarse en una carpeta dentro del repositorio GIT del grupo con el nombre **“Proyecto 1”**. Dentro de esta carpeta debe existir una carpeta con el nombre **“Entrega 1”** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto los archivos fuente de los diagramas como imágenes que se puedan leer con facilidad.
2. Entregue un enlace al repositorio a través de Brightspace en la actividad designada como **“Proyecto 1 - Entrega 1”**.

Entrega 2: Diseño

Teniendo en cuenta el análisis realizado en la primera entrega del proyecto, para la segunda entrega debe realizar el diseño detallado del sistema.

Actividades

1. Realice el diseño y construya un documento de diseño donde presente el diseño, haciendo especial énfasis en justificar las decisiones clave que haya tomado. El documento debe incluir, como mínimo, los siguientes elementos:
 - a. Un diagrama de clases de diseño que incluya todas las clases, incluyendo sus las relaciones, atributos y métodos.
 - b. Un diagrama de clases de alto nivel, que incluya todas las clases y sus relaciones, pero no todos los métodos ni atributos. Este diagrama facilitará entender las relaciones entre clases.
 - c. Diagramas de secuencia para las funcionalidades que usted considere críticas:Estos elementos NO son los únicos que debe incluir su documento: piense en qué otros diagramas, descripciones y justificaciones pueden ser necesarias para poder entender totalmente el diseño. Si su documento incluye únicamente los tres elementos mencionados antes, con seguridad será considerado insuficiente.

Entrega

1. Dentro de la carpeta del proyecto debe crear una carpeta con el nombre **“Entrega 2”** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto los archivos fuente de los diagramas como el documento de diseño.
2. Entregue un enlace al repositorio a través de Brightspace en la actividad designada como **“Proyecto 1 - Entrega 2”**.

Entrega 3: Diseño e Implementación

Teniendo en cuenta el análisis y el diseño, para la tercera entrega debe realizar construir la implementación del sistema.

Actividades

1. Si descubrió problemas con su diseño, actualícelo.
2. Implemente el sistema que diseñó. Tenga en cuenta que los detalles de la implementación deben ser coherentes tanto con el modelo de clases, como con los diagramas de secuencia.
3. Asegúrese de que la documentación del diseño sea consistente con la implementación.

Entrega

1. Dentro de la carpeta del proyecto debe crear una carpeta con el nombre “**Entrega 3**” donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto el documento de diseño actualizado como el proyecto Eclipse e instrucciones para su ejecución. Incluya también archivos de prueba para poder correr las aplicaciones y tener datos con los que se pueda probar con facilidad.
2. Entregue un enlace al repositorio a través de Brightspace en la actividad designada como “**Proyecto 1 - Entrega 3**”.