Documento de Análisis

da.acostac1

c.caro

j.serratos

Iteración 1:

Se determinó como rol el dominio general del proyecto y se definieron primeras responsabilidades

Roles: Seguimiento de proyectos

Responsabilidades:

1. Crear proyecto
2. Asignar dueño
3. Agregar participante
4. Registrar progreso en actividad
5. Crear actividad
6. Iniciar cronómetro
7. Finalizar cronómetro
8. Mostrar reporte de una actividad
9. Mostrar actividades de un usuario
10. Terminar proyecto

Iteración 2:

Se definieron conceptos que cumplen o son parte de las responsabilidades con sus debidos estereotipos y descripción

Proyecto: Controller

Es el encargado de crear y finalizar proyectos

Participante Inicial: Information Holder

Es el dueño del proyecto y almacena información de los miembros del proyecto y de él mismo

Participante: Information Holder

Es hijo de dueño, comparte sus atributos, guarda un historial de tiempos de trabajo, correo y nombre, al igual que la clase Participante Inicial

Registro: Information Holder

Es el encargado de almacenar información cuando una actividad de actualiza, por lo que es contenido por la clase Actividad

Actividad: Information Holder

Manipulador de actividades: Controller

Se encarga de actualizar los registros que hace un usuario sobre una actividad y tiene asociado los participantes de dicha actividad

Manipulador de usuarios: Controller

Se encarga de actualizar el status de los usuarios y actualizar los tiempos que lleva cada usuario en una actividad

Reflexión:

Ventajas:

Se añadió la clase participante como hijo de participante inicial ya que comparten los mismos atributos, lo cual es mejor que tener dos clases con sus propios atributos.

Desventajas:

Bastantes requerimientos no funcionales en los information holders aunque permita mostrar más información.

La comunicación entre controller e information holder es vital, no hay un delegador. Es decir, tiene un estilo de control centralizado cuando podría ser más distribuido.

Trade-offs:

Proyecto centralizado, es decir, depende de los information holder, pues la información se maneja desde los controller y estos deben estar conectados con los information holder. El encapsulamiento es alto respetando los atributos de cada clase y métodos de uso general, en conjunto, el acoplamiento es medio ya que no hay muchas dependencias de atributos de una clase sobre otras.

Iteración 3:

1. Crear proyecto -> Proyecto
2. Asignar dueño -> Manipulador de actividades
3. Agregar participante -> Manipulador de actividades
4. Registrar progreso en actividad -> Manipulador de actividades
5. Crear actividad -> Manipulador de actividades
6. Iniciar cronómetro -> Manipulador de actividades
7. Finalizar cronómetro -> Manipulador de actividades
8. Mostrar reporte de una actividad -> Manipulador de actividades
9. Mostrar actividades de un usuario -> Manipulador de usuarios
10. Terminar proyecto -> Proyecto

Restricciones:

* Un proyecto debe tener 1 solo participante inicial
* Una actividad solo tiene 1 participante
* La fecha de inicio / fin de una actividad no puede ser menor a la fecha de creación del proyecto
* La fecha de inicio / fin de una actividad no puede ser mayor a la fecha de finalización del proyecto
* La fecha de finalización de una actividad no puede ser menor a la fecha de creación de la misma
* La fecha de finalización de un proyecto no puede ser menor a la fecha de creación del mismo
* Solo el usuario principal puede terminar un proyecto

Diagrama de Clases

Diagrama

Descripción generada automáticamente

Diagrama de Análisis

Diagrama

Descripción generada automáticamente

Diagrama de Diseño

Diagrama

Descripción generada automáticamente

Diagramas de Secuencia:

