

Documento de Diseño Proyecto

1. Iteración

a. Roles

- i. Aplicación<<controller>>



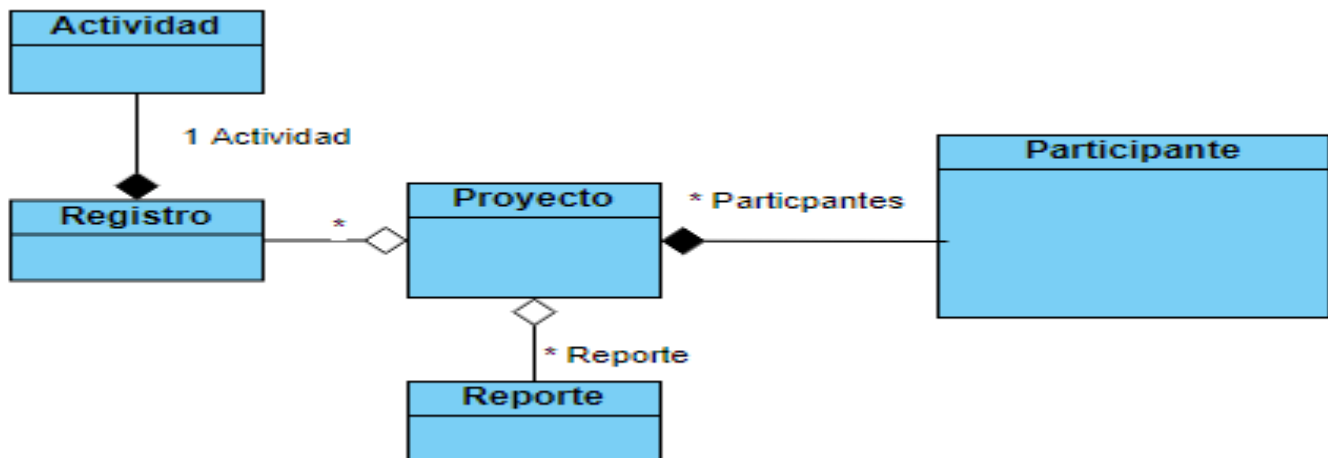
b. Responsabilidades

Responsabilidad	Rol
Correr aplicación	aplicación

2. Iteración

a. Roles

- i. Proyecto<<structurer>>
ii. Participante<<information holder>>
iii. Actividad<<structurer>>
iv. Registro<<info holder>>
v. Reporte<<structurer>>



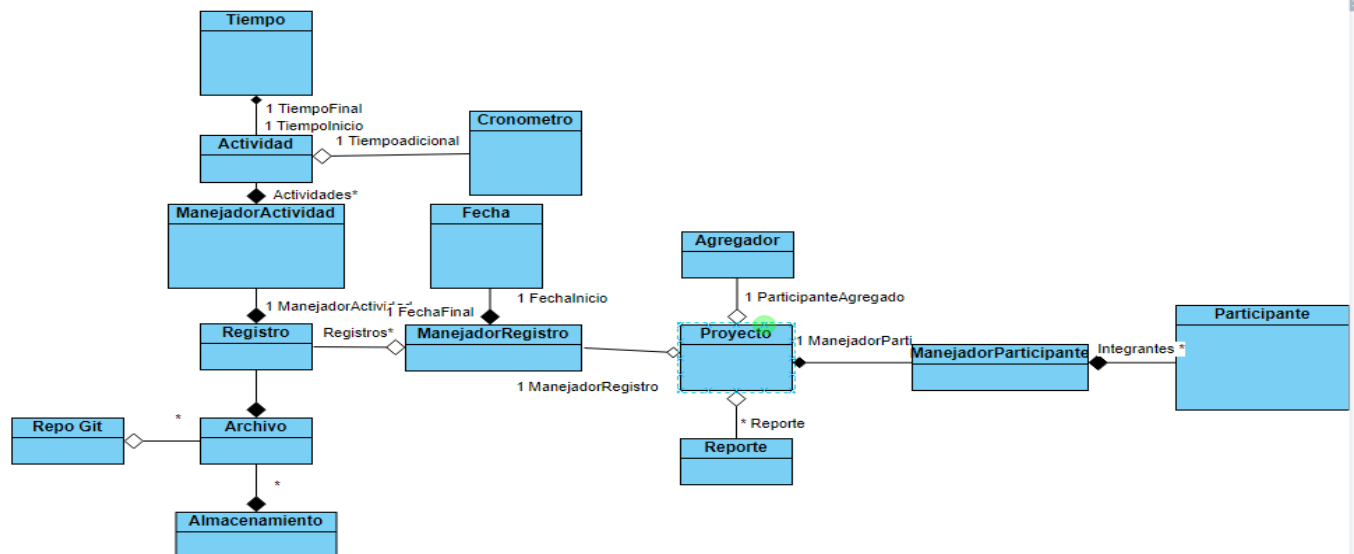
b. Responsabilidades

Responsabilidad	Rol
Crear proyecto	proyecto
Agregar participante	participante
Registrar actividad	participante
Mostrar reporte	reporte
Tomar tiempo	actividad
Modificar datos	participante
Mirar reporte	participante

3. Iteración

a. Roles

- i. Proyecto<<structurer>>
- ii. Participante<<information holder>>
- iii. Actividad<<structurer>>
- iv. Registro<<info holder>>
- v. Reporte<<structurer>>
- vi. Repo git/service
 - i. Archivo/info holder
 - ii. Almacenamiento/actividades
 - iii. Fecha/info
 - iv. Cronometro/ service
 - v. Agregador participante/ strcuturer
 - vi. Tiempo/ info holder
- vii. Manejadoractividades/structurer
- viii. Mnaejadoregistro/strcuturer
- ix. Manejadorparticipantes/strcutrer



b. Responsabilidades

Responsabilidad	Rol
Crear proyecto	proyecto
Agregar participante	participante
Registrar actividad	participante
Mostrar reporte	reporte
Tomar tiempo	actividad
Modificar datos	participante
Mirar reporte	participante
Manejar versiones	Repo git
Escribir actividad	archivo
Almacenar actividad	almacenamiento actividades

Registrar fecha inicio	fecha
Registrar fecha final	fecha
Pausar tiempo	cronometro
Agregar participante	agregador psrticipante
Tomar tiempo actividad	tiempo
Verificar info actividades	manejador actividades
Verficiar registros	manejador registro
Mirar participantes	manejador participantes

4. Iteración

a. Roles

- i. Proyecto<<structurer>>
- ii. Participante<<information holder>>
- iii. Actividad<<structurer>>
- iv. Registro<<info holder>>
- v. Reporte<<structurer>>
- vi. Repo git/service
 - x. Archivo/info holder
 - xi. Almacenamiento/actividades
 - xii. Fecha/info
 - xiii. Cronometro/ service
 - xiv. Agregador participante/ strcuturer
 - xv. Tiempo/ info holder
 - xvi. Manejadoractividades/structurer
 - xvii. Mnaejadoregistro/strcuturer
 - xviii. Manejadorparticipantes/strcutrer
 - i. Manejador version/ strcuturer
 - ii. Registraractividad/ info holder
 - iii. Tipoactividad/ info holder
 - iv. Manejadorarchivos/ strcuturer
 - v. Participante inicial/ info holder

b. Responsabilidades

Responsabilidad	Rol
Crear proyecto	proyecto
Agregar participante	participante
Registrar actividad	participante
Mostrar reporte	reporte
Tomar tiempo	actividad
Modificar datos	participante
Mirar reporte	participante
Manejar versiones	Repo git
Escribir actividad	archivo
Almacenar actividad	almacenamiento actividades
Registrar fecha inicio	fecha
Registrar fecha final	fecha
Pausar tiempo	cronometro
Agregar participante	agregador psrticipante
Tomar tiempo actividad	tiempo

Verificar info actividades	manejador actividades
Verficiar registros	manejador registro
Mirar participantes	manejador participantes
Verificar versiones	manejador versiones
Registrar nombre	registrar actividad
Definie tipo	tipoactividad
Veficiar archivo plano	manejador archivos
Crear proyecto	participante inicial

El diagrama final está en el pdf del repo Entrega 2

GLOSARIO

- c.** Manejador versión: Esta clase se va a encargar de manejar las versiones que se van a mandar al repositorio. En este caso, verificaría que sean archivos de texto planos.
- d.** Registrar actividad: Esta clase hace referencia a la que se va a encargar de registrar el usuario y la actividad que se este realizando en ese momento.
- e.** Tipo actividad: Esta clase hace referencia a la hora de definir los tipos de cada actividad.
- f.** Manejador archivos: Esta clase como su nombre lo indica se va a encargar de verificar los archivos que de van a almacenar de registros.
- g.** Participante inicial: Esta clase va a ser una herencia, de participante. Y su función es ser el dueño del proyecto, es decir, de poder agregar participantes.
- h.** Repo git/service: Esta clase hace referencia al servicio externo del proyecto. En el cual se van a almacenar las versiones del mismo.
- i.** Archivo: Esta clase es un texto el cual es el que se va a almacenar y a subir para tener el registro.
- j.** Almacenamiento: Esta clase se va a encargar de almacenar los archivos en el dispositivo, en el cual se esta trabajando.
- k.** Fecha: Esta clase es una extensión para predefinir las fechas y tener un formato cuando se pida.
- l.** Cronometro: Esta clase hace referencia al cronometro que se va llevar y mostrar al usuario para que este pause o reinicie su medida en la actividad.
- m.** Agregado participante: Esta clase hace referencia a lo que se necesita para poder agregar un participante. Es decir, nombre y correo y soltar un participante.
- n.** Tiempo: Esta es una clase la cual hace referencia al tiempo promedio y los diferentes tiempos que se le dan al usuario por actividad y por reporte.
- o.** Manejador actividades: Esta clase se va a encargar de administrar todas las actividades, verificar si se hicieron actividades en varios días por decir un ejemplo.
- p.** Manejador registro: Esta clase se encarga de administrar los registros. Por ejemplo: organizarlos, o ordenarlos.
- q.** Manejador participante: Esta clase es la que se encarga de llevar una administración de todos los participantes que están el proyecto.
- r.** Proyecto: Es el contenedor o la pieza principal de nuestra aplicación, en el van a estar todas las interacciones más importantes.
- s.** Participante: Esta clase hace referencia a los objetos que van a participar en el proyecto y puede ser tomada como interface.
- t.** Actividad: Esta clase va a ser una unión de varis y va a hacer la que haga la interacción con el sistema.

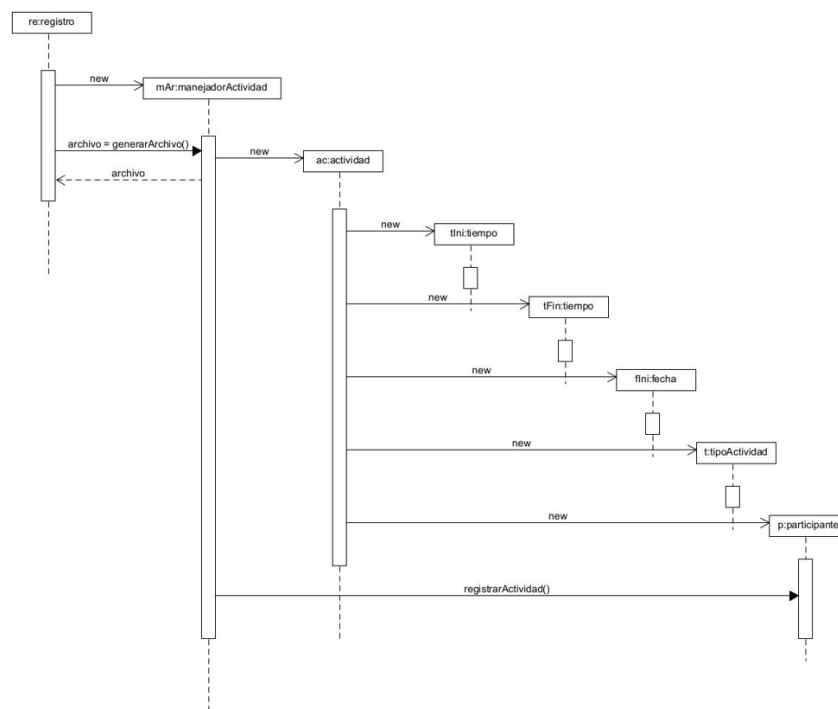
- u. Registro: Esta clase hace referencia a la información que se va a guardar en los archivos y registros. Además, de ser usada para estadísticas internas
- v. Reporte: Es el registro que se genera de cada actividad y es el que se va manejar en otros lados. Además, de tener la posibilidad de visualizarlo.

Estilo de control

El estilo de control que se lleva a cabo en el proyecto es uno delegado, esto se debe a la forma en la que están estructuradas las clases y como se dividen las responsabilidades de las mismas en general existirían 3 coordinadores: proyecto, actividad y archivo, estos principalmente dividen las actividades entre ellos mismo y las otras clases con las cuales tienen relaciones, un ejemplo de este seria proyecto el cual usa reporte para tener la información de los reportes, el agregador para agregar participantes y delegarle este elemento a esta clase y por ultimo también esta manejadorParticipante la cual se encarga de administrar a los participantes.

c. Colaboraciones

Caso 1: añadir actividad de un participante



Caso 2: cronometrar actividad de participante agregado

